AUGMENTED LINEAR INVERTED PENDULUM MODEL FOR BIPEDAL GAIT PLANNING

DAU VAN HUAN

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgments

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

I would like to express my sincere gratitude to my supervisor, Professor Poo Aun Neow, for his invaluable guidance, insightful advices, strong encouragements and generous support both academically and otherwise throughout the course of my PhD study. I also would like to thank my co-supervisor, Associate Professor Chew Chee Meng, for his supervision, helpful comments and full support for my study. His timely and visionary advices and feedbacks really helped to solve my problems and put me on the right track. I wish to thank my thesis committee members (Assoc. Professor Marcelo Ang and Assoc. Professor Hong Geok-Soon) for their time reading my thesis and giving useful feedbacks and comments.

I gratefully acknowledge the financial support provided by the National University of Singapore through Research Scholarship that makes it possible for me to pursue my PhD study. I am also grateful to the country of Singapore for giving me a great chance to study and live in Singapore.

Thanks are also given to my labmates (Weiwei, Albertus, Thuy, Wu ning, Dung, Tomasz, James and others) and technicians in Control and Mechatronics Lab for their support and encouragement. Thanks Weiwei and Albertus for your great friendship and fruitful discussions and comments on my research. Thanks my Vietnamese friends in NUS (Phuong, Hieu, Van, Trong, Huynh, Thanh, Dung, Phuoc, Nhu, Tho, Diem-Thanh, Chi) for their support and great friendship.

Finally, my thanks go to my parents and my brothers (Hoan and Hoang) for their continuous encouragements, moral supports and unconditional loves. Without them I would not have overcome the toughest times.

Table of Contents

Abstract

This thesis proposes a new model called the Augmented Linear Inverted Pendulum (ALIP) for bipedal walking. In this model, an augmented function F is added to the dynamic equation of the Linear Inverted Pendulum. The role of the augmented function is to improve the inverted pendulum dynamics by indirectly incorporating the dynamics of the arms, legs, heads, etc into the dynamics equation. The inverted pendulum dynamics can be easily adjusted or modified by changing the key parameters of the augmented function. Genetic algorithm is used to find the optimal value of the key parameters of the augmented function. Our objective is to design a walking pattern that has the highest stability margin possible.

The proposed ALIP model was used to generate off-line walking pattern for biped robot in 2D and 3D walking. Simulation results show that the proposed ALIP model is able to generate highly stable walking patterns. The walking patterns generated using the proposed approach is more stable than that generated using the LIPM model and GCIPM (an improved version of the LIPM model) model.

The ankle control strategy was proposed to improve stability margin. In this strategy, the ankle joint is controlled such that the ZMP stays as close to the middle point of the supporting foot as possible. This is obtained by adjusting the ankle pitch and roll angles based on the ground reaction force information so that the difference between the ground reaction force at the heel and toe is minimized. Simulation results show that the proposed method is effective in increasing the stability margin of the bipedal walking robot.

The proposed ALIP model was also successfully applied to generate online walking motion in sagittal plane. The online walking algorithm comprises of a proposed function called the Foot Placement Indicator (FPI). The Foot Placement Indicator (FPI) is an important part of the online walking algorithm. The role of the FPI is to decide the next walking steps (how far and how fast to take the next step) during the walking process based on the current states of the biped robot. Simulation results show that the obtained online walking motion is highly stable with large stability margin. In addition, the proposed algorithm is able to compensate for fairly large external disturbances affecting the walking robot.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Bipedal Locomotion

1.1.1 Definition

Bipedal locomotion is a form of movement where an object (human beings, animals, machines) moves by means of its two rear/lower limbs, or legs. An object that usually moves in a bipedal manner is usually known as **biped**.

There are four types of movements in bipedal locomotion including: Standing, Walking, Running and Jumping (or Hopping).

• Standing: Staying still on both legs. For human beings and animals the knees are locked while standing to minimized active control efforts. However, for humanoid robots the knee joints are usually powered in order to keep this standing posture.

• Walking: a process where the two feet exchanges support. One foot is in front of another with at least one foot on the ground at any time.

• Running: a process of feet exchange where one foot in front of another and there are periods where both feet are off the ground.

• Jumping/Hopping: a process where both legs are contracted and extended to generate reaction force that moves the object in a desired direction.

Among the four types of movements, probably walking is the most commonly used one. In this thesis we only focus on the walking motion. Therefore, from now on, bipedal locomotion is also referred to as bipedal walking.

1.1.2 Why Study Bipedal Locomotion?

Bipedal locomotion has been a topic of great interest of researchers for many years. There are plenty of reasons why we should study bipedal locomotion. Probably the key reason is that human beings have always dreamed of building machines that are similar to themselves (human beings are also bipedal). Since the ancient times, many people tried to design human-like machines. In 1206, Al-Jazari created hand washing automata with automatic humanoid servants [68], and an elephant clock incorporating an automatic humanoid mahout striking a cymbal on the half-hour. In 1495, Leonardo Da Vinci designed a humanoid automaton that looks like an armored knight, known as Leonardo's robot.

Another reason to study bipedal robots is because of their ability to navigate in rugged terrains where wheeled robots can not operate. Bipedal locomotion probably takes the smallest space compared to other types of robots. They can access areas where other robots can not such as staircases, stepping stones, or very narrow paths. Ideally, they can operate in a complex environment where human beings live and work. In addition, bipedal robots can be used in hostile or hazardous places where human beings can not work in.

Studying bipedal locomotion also gives us insights on how human beings walk [67]. In the past, when bipedal robots were not available, studies on human walking was done solely by biomechanics researchers and these research were carried out on human subjects. Nowadays, when supporting technologies for building bipedal robots is well developed and many advanced bipedal platforms have been built we can conduct research on human walking using these bipedal platforms. Although there are still differences in physical structure between bipedal robots and human beings, the basic walking gaits are similar. Doing research on bipedal robots one can test out different walking behaviors and scenarios where can not be done on human beings because of potential danger. Better understanding of bipedal locomotion would assist us in developing better leg prostheses for disabled people.

1.1.3 Challenges

Bipedal walking is a challenging control problem because it is a highly non-linear dynamics system [76]. Dealing with non-linear dynamics is always a difficult problem because it is very complicated and usually very hard to find analytical solution to these problems. Bipedal walking is also a multi-variable and naturally unstable dynamics.

Another characteristic that makes bipedal walking difficult is limited foot-ground contact[61]. This is a distinctive nature that makes it different from the control of robotics arms. Since the feet is not fixed to the ground as in robotic arms, it is likely that the supporting foot/feet would rotate over and cause the robot to fall if too much torque is applied at the ankle. This means that only limited control action can be applied during walking motion. The motion of other body parts such as arms, head and trunk must be properly planned so that they would not caused the robot to deviate from desired trajectories which may lead to a fall.

Bipedal walking robot is a discretely changing dynamics system. During a walking cycle, the exchange in feet/foot support causes a change in robot's dynamics. Due to the non-continuous property of the dynamics equations, it is challenging to apply traditional techniques to stabilize the system.

Stability is a critical issue of bipedal walking but there is still no clear and unified criterion of stability so far. One of the commonly used stability criterions in bipedal walking is the Zero-moment-point [83, 82, 84]. However, this method also has its own disadvantages as it can not guarantee stability in some special cases.

1.2 Motivation

Bipedal walking research has been around for over thirty years. Many research works have been introduced to tackle the challenges of bipedal walking. These works can be classified into the following methods: Model-based approach, ZMP-based approach, Learning-based approach, Central Pattern Generator approach, and Angular Momentum-based approach.

Among these approaches, model-based approach seems to be the most comprehensive and straightforward approach to bipedal gait planning. Model-based approach is an approach whereby dynamics of the physical robot is modeled using mathematical representation. The mathematical representation is also referred to as the dynamic equation of the system. In order to analyze the system dynamics, it is usually required to solve the dynamics equation for interested parameters. Once the solution is known, it is straightforward to plan the walking gait for bipedal robots. However, due to the high level of complexity and non-linearity of bipedal walking dynamics, it's almost impossible to find analytical solution for the complete dynamics bipedal model. Therefore, many researchers choose to simplify the dynamics model so that analytical solution can be obtained.

There are two ways to simplify the dynamics equation. The first way is to linearize the dynamics equation of complex model at equilibrium points. The advantage of this method is that the analytical solution can be obtained without having to change the original complex dynamic model. However, this method only works well in a limited range around the equilibrium point. When the state of the system is far from the equilibrium point, the solution is not effective anymore. This is not desirable because it reduces the flexibility of the algorithm. The second way to simplify the dynamics equation is to use simpler dynamics models. This is done by neglecting the inertia properties, joint friction, actuator dynamics of some parts of the robot such as legs or arms. One good example of this method is the Linear Inverted Pendulum model [38, 37, 40]. In this model, the dynamics of bipedal walking robot is modeled as one point mass attached to the tip of the inverted pendulum. The dynamics of arms and legs are ignored in this model. The mathematical representation of this dynamic model is very simple and it is easy and straightforward to find analytical solution for the dynamic equation. The advantage of this model is that analytical solution can be obtained easily and this solution is a general solution applicable to any state of the robot. However, since this model is too simple, it may not be easy to control the robot to follow the desired reference trajectory generated using this approach if the difference between this model and the actual physical robot is too big.

In view of the above analysis, the Model-based approach would be an excellent and promising approach if one could find a simple dynamic model yet be able to take into account (directly or indirectly) more complete dynamic behaviors. If such a model is available then the gait planning task is simple and straightforward. Moreover, various dynamic behaviors can be derived and implemented easily based on this model. Therefore, in this thesis, we are strongly interested in finding such a model.

1.3 Objective and Scope

The objective of this thesis is to construct a comprehensive and effective method of walking gait synthesis for bipedal robot. The walking gait obtained by the proposed method must allow the robot to achieve a stable 3D dynamic walking and fulfill the following walking requirements:

• It should be applicable for real-time implementation. • It should be applicable to bipeds of different mass and length parameters. • It should be able to compensate for large external disturbances.

The scope of this thesis is restricted to bipedal walking on level ground along a straight path. The external disturbance caused by the unevenness of the terrain will not be considered. Instead, an external force is applied on the robot's body to test the effectiveness and robustness of the algorithm.

1.4 Approach

This section briefly explains how the motivation presented in the last section can be realized and implemented. It is desirable to have a dynamic model not too complex so that analytical solution can be obtained and at the same time not too simple so that some important dynamics will not be ignored.

In bipedal walking literature, the well known Linear Inverted Pendulum model proposed by Kajita *et al.*[39, 34, 37] is a simple and effective model to describe bipedal walking motion. The dynamic equation of this model can be solved analytically without using any linearization technique. This model provides useful dynamic insights which are vital for planning bipedal walking gaits. However, since the Linear Inverted Pendulum model is a very simplified model of bipedal walking robot, the desired walking gait generated using this model may not be easy to realize if the difference between the dynamic model and the actual robot is significant.

In this thesis, a new model called the Augmented Linear Inverted Pendulum (ALIP) [11] is proposed. An augmented function F is added to the dynamic equation of the Linear Inverted Pendulum. The role of the augmented function is to improve the inverted pendulum dynamics such that the disturbance caused by the un-modeled dynamics (legs and arms, etc.) is minimized. The augmented function has two key parameters whose values are changeable. When the key parameters change, the dynamic equation changes accordingly. Genetic algorithm [17] is used to find the optimal value of the key parameters. The objective of our proposed method is to achieve the highest stability margin for bipedal walking. It is noted that full dynamics of the robot is considered when computing the stability margin during the optimization process. Therefore, it is reasonable to say that the proposed ALIP model is closer to the actual physical model compared to the Linear Inverted Pendulum model because dynamics of arms and legs are indirectly considered through the use of the augmented function.

The proposed ALIP model is applied to plan the *offline* walking gait for humanoid robot. Both 2D and 3D walking are considered. To further enhance stability, the ankle control strategy is introduced. In this strategy, the ankle joints are used to adjust the feet angles to make sure the foot/feet is in full contact with the ground.

The proposed ALIP model is also successfully applied to generate *online* walking gaits. In this thesis, the online walking algorithm is based on the centre of mass (COM) velocity information. This is because velocity is one of the most important factors determining the stability of bipedal walking. Indeed, the magnitude of the COM velocity would determine how far and how fast the swing leg must swing in order for the robot to stay balanced and maintain desired walking speed. When the step length is constant, the higher the walking speed, the smaller the step time (the swing leg must swing faster) and vice versa. When the step time is constant, the higher the walking speed, the larger the step length the robot must take to capture balance. To test the effectiveness of the algorithm, disturbance force is exerted on the robot during the walking process.

1.5 Targeted Biped Robot

This section describes the bipedal robot used to test our proposed method. The robot's name is HUBIRO (see Figure 1.1). The robot's height is $1.7m$, total weight $86.59kg$. HUBIRO has a total of 28 degrees of freedom (DOF) of which 6 DOFs at each leg (Hip Pitch, Hip Roll, Hip Yaw, Knee Pitch, Ankle Pitch, Ankle Roll), 6 DOFs at each arm, 2 DOFs at the waist and 2 DOFs at the neck. The biped has a rotary angular position sensor at each DOF. In the simulation model, it is assumed that the robot has a single axis gyroscope fixed to the body which can provide the body posture information (body

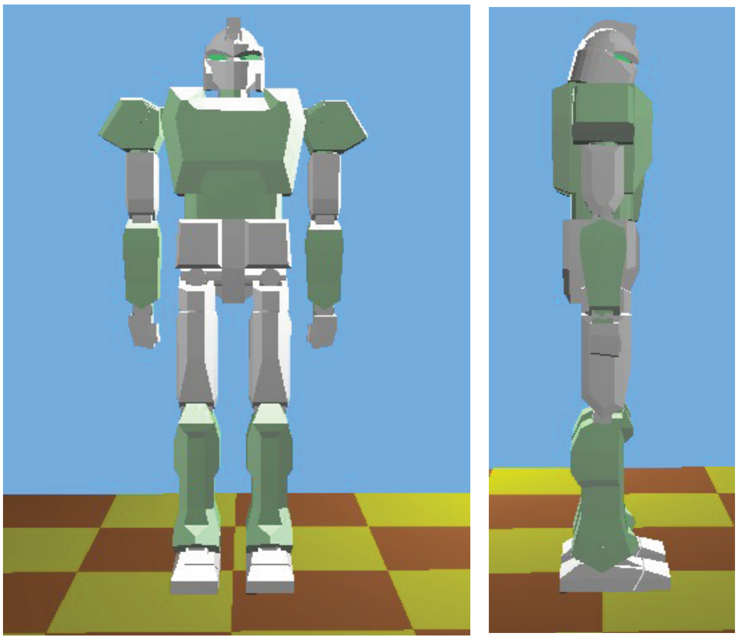


Figure 1.1: *Picture of HUBIRO*

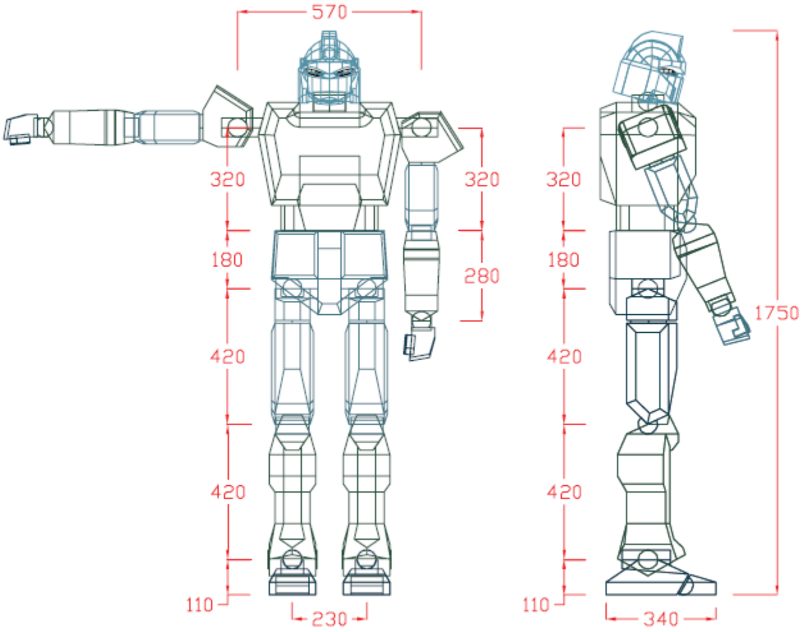


Figure 1.2: *Basic dimensions of HUBIRO*

Table 1.1: Specifications of HUBIRO.

Description	Value
Total Mass	86.59 kg
Body Mass	54.49kg
Thigh Mass	4.69kg
Shank Mass	8.63kg
Foot Mass	2.73kg
Body's Principle Moment of Inertia	
x-axis	0.89kgm^2
y-axis	0.46kgm^2
z-axis	1.25kgm^2
Thigh's Principle Moment of Inertia	
x-axis	0.19kgm^2
y-axis	0.02kgm^2
z-axis	0.19kgm^2
Shank's Principle Moment of Inertia	
x-axis	0.95kgm^2
y-axis	0.03kgm^2
z-axis	0.95kgm^2
Foot's Principle Moment of Inertia	
x-axis	0.02kgm^2
y-axis	0.01kgm^2
z-axis	0.02kgm^2
Hip Spacing	0.23m
Thigh Length	0.42m
Shank Length	0.42m
Ankle Height	0.11m
Foot Length	0.34m
Foot Width	0.144m

pitch, roll, yaw angles). Each foot has four force sensors at the bottom to measure the ground reaction forces on the robot. The joints of the robot are driven by electrical brushless DC motors.

The basic dimensions of the robot are shown in Figure 1.2. The specifications of the robot is presented in Table 1.1.

1.6 Simulation Tools

In this thesis, two dynamic simulation software are used. The first one is Yobotics Simulation Construction Set or Yobotics in short (http:www.yobotics.com) developed by Yobotics Inc. and the second one is Webots developed by Cyberbotics Ltd. Yobotics is used to simulate 2D walking motion while Webots is used to simulate 3D walking. The reason for this is because Yobotics supports 2D simulation (Webots doesn't) and Webots is better compared to Yobotics in terms of 3D simulation.

1.6.1 Yobotics

The Yobotics Simulation Construction Set is a full-featured software package for easily and quickly creating simulations of robots, bio-mechanical systems, and mechanical devices. The Simulation Construction Set is easy to use, yet powerful for creating complex simulations of robotic devices. Arbitrary control can be added to these devices as each degree of freedom automatically has a simulated actuator associated with it.

The dynamic interaction between the biped and the terrain is established by specifying four ground contact points (two at the heal and two at the toe) beneath each of the feet. The ground contacts are modeled 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 correct deflection of the ground contact model.

One good point about Yobotics is that it allows user to simulate dynamics in 2D space. The dynamics is activated on the Sagittal plane and freezed on the Frontal plane. In this thesis, we will use Yobotics to simulate 2D walking. This is recommendable because doing simulation in 2D is much simpler compared to 3D simulation yet it still help us to test the effectiveness of the algorithm. We only move on to 3D simulation when the 2D one works well as expected. Doing this can save us a lot of time.

1.6.2 Webots

Webots is a professional mobile robot simulation software package. It offers a rapid prototyping environment, that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc. The user can add simple passive objects or active objects called mobile robots. These robots can have different locomotion schemes (wheeled robots, legged robots, or flying robots) [85].

Webots simulation engine uses virtual time, thus making it possible to run a simulation often much faster than real robots. Webots utilizes the Open Dynamics Engine (ODE), a powerful tool, to perform accurate physical simulation.

A great advantage of Webots is that it allows users to specify the bounding objects for collision detection. The contact surface of the foot/feet can be represented by a bounding box. Therefore, the contact between the foot and the ground is a surface contact, a more realistic contact compared to the four-point-contact used in Yobotics.

In this thesis, we will use Webots for 3D simulation tasks.

1.7 Contributions of this PhD thesis

The contributions of this thesis are:

(1) The proposal of a new dynamic model for bipedal walking called the Augmented Linear Inverted Pendulum (ALIP).

(2) The application of the proposed dynamic model ALIP for generating reference walking patterns for bipedal robots. The reference walking patterns are applied in both 2D and 3D walking experiments.

(3) The analysis of the effect of the speed and mass of the swing leg on the stability of bipedal walking robots.

(4) The demonstration of the generality of the proposed walking algorithm when different specifications of the bipedal robots are used.

(5) The development of an online walking algorithm that can adapt well with the changes in walking speed and external disturbances.

(6) The demonstration of the effectiveness of the ankle strategy used to increase stability for bipedal walking.

1.8 Thesis Outline

Chapter two presents the literature review of the bipedal walking research which is related to the work in this thesis. The bipedal walking research are classified into groups based on the approaches used.

Chapter three describes in details some simple dynamic models of bipedal walking such as Inverted Pendulum, Linear Inverted Pendulum, Gravity-compensated Inverted Pendulum. These models are closely related to the proposed approach in this thesis. Advantages and disadvantages of these approaches will also be discussed.

Chapter four presents in details the formulation of the proposed model called the Augmented Linear Inverted Pendulum (ALIP). The proposal of the ALIP model is one of the most important contributions in this thesis. This model is used to generate walking gaits for humanoid robots.

Chapter five shows how the proposed model ALIP can be applied for generating 2D offline walking patterns. In this chapter, the computation of the Zero-moment-point (ZMP), an important stability criterion in bipedal walking, is also presented. In addition, the application of the genetic algorithm to find optimal value of key parameters is also mentioned in details. To improve stability margin, the ankle pitch strategy was adopted. Finally, some simulation results are reported to prove the effectiveness of the proposed method.

Chapter six describes the application of the proposed ALIP model in frontal plane and the extension to 3D walking gait generation. This chapter also illustrates how one could utilize the ankle roll joints to control the ZMP so that better stability can be achieved.

Chapter seven illustrates the application of the proposed ALIP model to generate online walking gaits in sagittal plane. The Foot Placement Indicator (FPI), an important part of the algorithm, is presented in details. The effectiveness of the proposed algorithm is proved through two walking simulations (with and without external disturbances).

Chapter eight concludes this thesis and proposes some possible directions for future studies.

Chapter 2

Literature Review

Bipedal walking research has been around for decades. Inspired by the success of building autonomous bipedal walking machines, more and more walking algorithms have been developed and improved by researchers around the globe. Bipedal walking algorithms can be classified into the following methods:

- Model-based Method
- ZMP-based Method
- Learning-based Method
- Central Pattern Generator Method
- Passive Dynamics Walking Method
- Angular-Momentum-based Method

In the following subsections, brief reviews of each method will be presented.

2.1 Model-based Method

In this approach, mathematical model of the biped robot is extracted from its intrinsic dynamic characteristics. Due to the complexity and non-linearity of biped models, most of the dynamic equations of these models do not have analytical solutions. Therefore, many researchers tend to simplify biped models to simpler ones such as inverted pendulum, linear inverted pendulum, double-pendulum model, etc. which are massless-leg models neglecting inertia properties of the legs, joint friction, link flexibility, actuator dynamics, etc. Whereas others try to linearize dynamic equations of the complex models at equilibrium points so that analytical solutions could be achieved.

Kajita *et al.* [40] derived an ideal massless-leg biped model called the Linear Inverted Pendulum Mode (LIPM). In this model, the center of gravity (COG) of the body moves horizontally and the horizontal motion of the COG can be expressed by a simple linear differential equation. They introduced the term ”potential energy conserving orbit” to describe this class of trajectories. The obtained dynamic equation has analytical solution which can be used directly for walking gait planning. To make the walking motion robust, they proposed attitude control using local feedback and adaptive support leg exchange. This approach was successfully applied to bipedal robots walking on rugged terrain [37], and 3D walking pattern generation [34, 31].

Park *et al.* [56] improved the LIPM by introducing the term ”Gravity-Compensated

Inverted Pendulum Mode" (GCIPM). Their approach takes into account the gravity of the swing leg to generate biped locomotion patterns. The walking trajectory generated by this approach is more stable than the one generated by the LIPM method.

Suzuki *et al.* [77] proposed a trajectory planning method which incorporates two kinds of inverted pendulum which are the linear inverted pendulum (LIPM) and the normal inverted pendulum with constant leg length (IPM-C). The switching mode is necessary to switch the control between LIPM and IPM-C. The simulation shows that this method is more efficient compared to the method using only one type of inverted pendulum.

The great advantage of these inverted pendulum-based methods is that the dynamic model is simple hence it is easy to get analytical solution. It is quite easy and straightforward to design the trajectory once the analytical solution is obtained. However, since the model is too simple it may cause problem when controlling the real biped due to the significant difference between the model and the real robot. Additional control strategies are usually adopted to make the walking possible.

When leg's inertia is not negligible, it needs to be included in the biped model. Acrobot model [74, 51] is one of the famous models that includes the leg's inertia. It is a double pendulum model without actuation between the ground and the base link. The Acrobot dynamics are complex enough to yield a rich source of nonlinear control problem, yet simple enough to permit a complete mathematical analysis.

When more dynamics of the robot such as leg's inertia, joint friction, actuator, etc are taken into account, the overall dynamic equations become very nonlinear and complicated. To deal with this problem, linearization approach is usually adopted to simplify these dynamic equations.

Miura *et al.* [47] built a 3D walking biped that had three links and three actuated degrees of freedom: one at each of the hip roll joints and one for fore and aft motion of the legs. The ankle joint were limp. In order to design walking controller, the authors proposed a linearized dynamic model with the assumption that the motions about the roll, pitch and yaw axes were independent. And the yaw motion was assumed negligible. The state feedback control laws were formulated after selecting a set of feasible trajectories for the joints. The control laws generate compensating inputs for the reference control inputs and ensured the convergence of the actual trajectories to the desired trajectories. One disadvantage of this method is that the motion space had to be constrained to a smaller one because the linearized model is only valid in a limited range.

2.2 ZMP-based Method

The concept of ZMP (Zero Moment Point) was first introduced by Vukobratovic *et al.* [83] in 1970. ZMP is a stability index of dynamic walking for biped robot. It is defined as the location on the ground where the total moment generated from the ground reaction forces has zero moment about two axes that lie in the plane of the ground. Takanishi et al. [80], Hirai *et al.* [22], Fujimoto *et al.* [16] proposed methods of walking pattern synthesis based on the ZMP, and demonstrate walking motion with real robots. Basically, these approaches first design a desired ZMP trajectory, then derive the torso motion to realize the desired ZMP trajectory. However, since the change of the ZMP due to the torso motion is limited, not all desired ZMP trajectories will be realizable.

Huang *et al.* [24] proposed a method to plan a walking pattern consisting of a foot trajectory and a hip trajectory. The ZMP was used as an index to evaluate stability of the biped robot. The complete foot trajectory was generated using a third-order spline interpolation which taken into account the foot constraints. Similarly, the hip trajectory was generated using a third-order spline function. Some of the key parameters of the hip trajectory was allowed to vary in a certain ranges. The optimal values of these parameters were optimized in such a way that the resulting trajectory has the highest stability margin.

Yang *et al.* [87, 88] proposed a method for bipedal walking gait generation based on a Truncated Fourier Series Formulation with coefficients tuned by Genetic Algorithm. This method allows the adjustment of the stride-frequency, step-length real-time. The ZMP is used as a stability criterion to ensure the feasibility of the planned walking gaits.

Shih *et al.* [70] proposed an approach optimizing the biped walking trajectory that can be used as a reference trajectory for control. The biped is modeled as a kinematic chain of 13 links connected by 12 joints. The inverse kinematics of the biped is derived for the specified positions of the body and feet. The aim is to optimize the location of the ZMP. Specifically, they minimize the deviation between the ZMP and the center of shape of the supporting area by considering the position of the body as free variables.

Park *et al.* [57] proposed a method to reduce the motion range of the trunk by generating a designed trajectory of the ZMP. The trajectory is determined by a fuzzy logic based on the leg trajectories that are arbitrarily selected. The resulting ZMP trajectory is similar to human's one and the ZMP continuously moves forward. The proposed approach is simulated on a 7-DOF biped robot. Simulation results show that the proposed ZMP trajectory increases the stability of the locomotion and thus resulting in reduction of motion range of the trunk.

Kajita *et al.* [33] presented a method called Preview Control of Zero-Moment-Point. In this method, the dynamics of a biped robot was modeled as a running cart on a table. With this model, the ZMP was easily treated. They adopted the preview control theory that uses the future references to design the ZMP tracking servo controller. It was shown that a preview controller can be used to compensate the ZMP error caused by the difference between a simple model and the precise multibody model.

ZMP is widely used as a criterion for dynamic stability of biped robot by many researchers. The good points about ZMP-based method are that it is a well defined methodology to prove stability and easy to implement in a real robot. However, it also has some disadvantages. Firstly, there is a need to be able to modify on-line the desired trajectory and this is not simple at all. Furthermore, the use of ZMP requires that the robot has feet (foot/feet is a physical structure that provides at least 2 contact points (2D) or 3 contact points (3D) between the robot's leg and the ground). In addition, this criterion is not applicable in the flight phases during the running motion.

2.3 Learning-based Method

Learning approach is usually used when little knowledge of the system is known or dynamic model of the system is too complicated. It is also used to modify nominal behaviors generated based on a simplified model.

Miller [45] designed a learning system using neural networks for a biped that was capable of learning the side-to-side and front-to-back motion. With the help of several control strategies, the neural networks learn to provide feedforward control to the joints. The biped had a slow walking gait after training.

Mori *et al.* [49] presented a reinforcement learning (RL) method for a central pattern generator (CPG) controller. In this study, they proposed a learning scheme for a CPG controller called a CPG actor-critic model, whose learning algorithm is based on a policy gradient method [42]. They applied their RL method to autonomous acquisition of biped locomotion by a biped robot simulator. Although the simulation was successful, a lot of training episodes were still required. Therefore, it is difficult to apply the method directly to real robots; it is necessary to develop a more efficient algorithm which enables the robot to learn faster.

Chew [6, 5] used Q-learning (one type of reinforcement learning) algorithm to learn the key parameters of the swing leg control task so that speed control can be achieved. In order to reduce the complexity of the walking system, the divide-and-conquer approach was used. 3D bipedal walking was broken down into motion controls in the transverse, sagittal and frontal planes. Each of these was then considered individually. By dividing the walking task into subtasks, the learning time can be significantly reduced. In addition, the local speed control based on the stance ankle was adopted to supplement the control algorithm. To globally control the walking speed, swing leg control strategies were used. This was done by learning the appropriate swing time and step length so that stable walking at desired speeds can be achieved. In this study, the CMAC (Cerebellar Model Articulation Controller) was used as a function approximator for Q-learning. By applying the mentioned control algorithm, 3D walking motion was successfully obtained in flat and uneven terrains.

Benbrahim and Franklin [3] proposed a control algorithm using a ”melting pot”. The melting pot is a central controller that uses the experience of other peripheral controller in order to learn an average control policy. The central controller was pre-trained to provide nominal trajectories to the joints. The peripheral controllers intervene only when they consider that the controller’s action violates their individual control policies. The central controller and some of the peripheral controllers in this study use adaptive CMAC (Cerebellar Model Articulation Controller) neural networks. The desired walking pattern is defined by a set of constraints. The robot learns to execute movements that satisfy all of these constraints. No dynamic model of the robot system was required in the implementation. The proposed approach has one disadvantage is that it is required to pre-train the central controller using the nominal joint trajectories. This task is usually not easily obtained.

Morimoto *et al.*[50] proposed a model-based reinforcement learning algorithm for biped walking in which the robot learns to appropriately place the swing leg. This decision is based on a learned model of the Poincare map of the periodic walking pattern. The model maps from a state at the middle of a step and foot placement to a state at next middle of a step. They also modified the desired walking cycle frequency based on online measurements. To estimate appropriate walking cycle timing they used phase oscillators.

The advantage of learning-based methods is that there is no need to derive and solve nonlinear, complicated dynamic equations of the robot systems. However, since not much dynamic information is available, the learning agents must spend a huge amount of trials and errors before achieving desired behaviors. A large memory is then required to store all the states. Therefore, the effectiveness of the method depends on how to decompose the control task and at which level of the control learning is applied. It is also dependent very much on the performance index chosen.

2.4 Central Pattern Generator

Biological bipeds such as birds, humans have excellent ability to walk with high stability, adaptability, and agility. Therefore, many researchers have been trying to explore biological bipeds in order to extract good algorithms applicable to bipedal robots [20, 19, 21, 66, 28, 65, 14]. Basically, the normal procedure comprises of the following steps. First, the behaviors of the biological system are observed and analyzed. Then, mathematical models and hypotheses for the behaviors are proposed. Lastly, those models are verified through simulations and/or experiments.

Grillner [69] found from experiments on cats that the signal controlling the muscles to perform coordinated walking motion was generated by the spinal cord. Based on this finding, he hypothesized a Central Pattern Generator (CPG) which is a network of neurons in the spinal cord.

The concept of CPG has inspired many researchers to adopt it for bipedal locomotion [79, 78, 64, 53, 52]. Most of the studies use a system of coupled nonlinear equations to generate signals for the joint trajectories of bipeds. Taga *et al.* [79] proposed a new principle of sensorimotor control of legged locomotion in an unpredictable environment on the basis of neurophysiological knowledge and a theory of nonlinear dynamics. Stable and flexible locomotion is realized as a global limit cycle generated by a global entrainment between the rhythmic activities of a nervous system composed of coupled neural oscillators and the rhythmic movements of a musculo-skeletal system. Coordinated movements were generated by dynamic interactions among the nervous system, the musculo-skeletal system and the environment.

Huang *et al.* [25, 26, 27] studied the coordination between neural oscillators in CPG and applied to bipedal walking gait planning. By using the entrainment property of the neural oscillator, they developed a method which uses the difference between oscillator's output and desired output to adjust the inner states of neural oscillators. A CPG structure with coordination between neural oscillators was proposed to control the bipedal robot. The robot can walk stably and robustly even under external disturbing forces acting on the robot during walking.

Nakanishi *et al.* [52] introduced a framework for learning biped locomotion using dynamical movement primitives based on non-linear oscillators. Their ultimate objective is to establish a design principle of a controller in order to achieve natural human-like locomotion. They suggested dynamical movement primitives as a central pattern generator (CPG) of a biped robot. Demonstrated trajectories were learned through movement primitives by locally weighted regression, and the frequency of the learned trajectories were adjusted automatically by a novel frequency adaptation algorithm based on phase resetting and entrainment of coupled oscillators.

The advantages of CPG approach are that it doesn't require exact knowledge of the robot's model, the CPG has the ability to recover smoothly after a perturbation. In addition, no complicated control input is required and the non-linear oscillators on which the CPG is based are simple and local elements. However, the CPG approach also has drawbacks. Firstly, there are many parameters to be tuned. Another disadvantage is that it is difficult to find a set of parameters that enable entrainment of the overall system. Furthermore, it is difficult to change the behavior or add more behaviors.

2.5 Passive Dynamics Walking

Passive dynamics was first introduced by Tad McGeer [43, 44] in 1990. It was inspired by a bipedal toy that had the ability of walking down a slope without using any actuator system. The toy rocked left and right in a periodic motion. When a leg is lifted in the process, it could swing freely forward 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 behavior can be achieved. When this happens, the work done on the toy by the gravitational force equals the energy loss by the biped.

Passive walking has one interesting advantage is that it can achieve a minimum energy gait without active control. However, it also has some disadvantages like the sensitivity to parameter variations [43], such as, the mass distribution, joint friction, etc. During physical implementation, iterative tuning is usually required before a successful implementation can be achieved. Moreover, it is still not clear how it can be implemented for other terrains like level ground, rugged surfaces, etc.

More and more research on passive dynamics walking are being done nowadays [41, 48, 55]. In the future, there should be more studies on how to utilize the advantages of passive dynamics to reduce energy consumption of actuated bipedal robots. It will be very effective since all biped robots operating on different terrains must have actuators. Then, the question is how to minimize the energy consumption. Utilizing passive dynamics in combination with other approaches may be a good choice.

2.6 Angular-Momentum-based Method

In recent years, some researchers turn their attention to angular momentum with the hope to find a more generalized stability criterion for bipedal walking [18, 32, 59, 60, 1].

Goswami *et al.* [18] proposed a new method to recapture balance of bipedal robots based on a fundamental principle of mechanics which states that the resultant external moment on a system, computed at its CoM, is equal to the rate of change of its centroidal angular momentum $\dot{H}_G$. A bipedal robot is rotationally stable if the external forces and moments sum up to a zero centroidal moment or the angular momentum of the system is conserved ($\dot{H}_G = 0$). Based on this idea, he proposed three control strategies to regain balance which are (1) enlarging the support polygon, (2) moving the centre of gravity G and (3) changing the ground reaction force (GRF) direction.

Kajita *et al.* [32] proposed a new control method called Resolved Momentum Control in which the total linear/angular momenta is first specified and then the whole body motion of the humanoid robot was derived to obtain that desired momenta. The whole body motion was calculated from a given momentum reference by using a pseudo-inverse of the inertia matrix. Using this method, they could generate the kicking and walking motions which were successfully tested on the actual humanoid robot HRP-2.

Motivated by biomechanical studies on human walking, Popovic *et al.* [59] hypothesized that spin angular momentum in human walking is highly regulated by the central nervous system ($\vec{S} \approx 0$), where $\vec{S}$ is the total angular momentum. They tested this hypothesis by computing the total spin angular momentum and the rotational effects of that momentum using kinematic gait data measured from a human test subject and a morphologically realistic human model. Based on this hypothesis, they derived the nonlinear coupling between the ground reaction force, center of mass position (CM) and center of pressure location (CP). Employing this relationship, they proposed a method to rapidly generate biologically realistic CP and CM reference trajectories.

Angular-Momentum-based approaches seems to work fairly well. However, walking with Minimized Spin Angular Momentum is not a necessary condition for stable walking. In fact, one can walk while freely changing his angular momentum by thrashing his upper body's mass around. Minimizing spin angular momentum is also not a sufficient condition for stable walking, as a biped robot may fall over while maintaining an angular momentum of zero.

2.7 Summary

In the above subsections, research on dynamic bipedal walking was classified into different groups. The classification was done based on the ways researchers solve gait planning problem. Researchers either relied on the dynamic model of the biped (Model-based), or learning and intuition (Learning-based), or stability criterions (Angular-momentum-based and Zero-moment-based) to design walking gait for biped.

The Model-based approach would be a comprehensive and excellent approach if one could find a not too complex model yet still be able to represent nonlinear dynamic behaviors. If such a model is available then the gait planning task is quite simple and straightforward. Moreover, various dynamic behaviors can be derived easily from the model. The problem is that a complete model is usually too complex to analyze and it's not easy to get exact model information such as mass distribution and inertia terms.

The ZMP-based approach is a well defined methodology and easy to implement in a real robot. However, it also has disadvantages. This method requires that the biped robot must have feet. This is quite limited because there are robots without feet but still can walk stably. Furthermore, this approach can not be applied to running or jumping motions.

The Learning-based approach is interesting because stable walking motion can be obtained without the need to derive and solve complex dynamic equations. However, intuition is still required to speed up the learning rates. Also, choosing of performance measure for a good walking behavior can be tricky sometimes. Besides, learning approach can become intractable if there are too many learning agents.

The Central Pattern Generator (CPG) approach is good at generating periodic walking patterns. However, there is no good reason why the walking gait must be periodical. There are times when human walking is not periodic such as walking over obstacles or walking on random stepping stones. The disadvantage of this method is that there are many parameters to be tuned and it is difficult to find a set of parameters that enable entrainment of the overall system.

The Angular Momentum-based method can be a good method to generate human-like walking gaits (in terms of angular momentum regulation). However, it is not quite clear how minimizing angular momentum related with stable walking. This question need to be explored further. The good thing about minimizing angular momentum is that it gives a reserve in spin angular momentum which can be utilized to help recover from a push or other disturbances [62].

Chapter 3

Simple Models of Bipedal Walking

This chapter presents the analysis of some simple models of bipedal walking including the Linear Inverted Pendulum Model (LIPM) and the Gravity-compensated Inverted Pendulum Model (GCIPM). The advantages and disadvantages of these methods will be discussed. In addition, in this chapter, the effect of the swing leg's dynamics on the robot's dynamics is studied. Results show that when the swing leg's mass is big enough (say more than 10% of the body mass), the effect of the swing leg's dynamics can not be ignored. Or when the swing leg's speed is high, the swing leg's dynamics has significant effect on the whole robot dynamics.

3.1 Linear Inverted Pendulum Model (LIPM)

The Linear Inverted Pendulum Model (LIPM) was adopted by researchers [13, 36, 35, 15, 63, 11, 37, 38, 40, 30] to design bipedal walking gaits. In this model, the total mass of the robot is assumed to be concentrated at one point called the Center of Mass (CoM). The CoM's height is assumed to follow a linear path by controlling the knee joint. The legs are assumed massless. For level ground walking, the height is kept constant. The length of the beam representing the leg is variable. The Linear Inverted Pendulum Model is shown as in Fig. 3.1. Assuming that the ankle torque is zero, the dynamic equation of the model is as follows:

$$\ddot{x} = \frac{g}{z_o}x \tag{3.1}$$

where x is the horizontal coordinate of the point mass from the vertical plane that passes through the ankle joint; z_0 is the body's constant height; g is the gravitational constant (see Fig. 3.1).

If the initial horizontal position and velocity x_i and $\dot{x}_i$ are given, the trajectory of the body's CoM is presented by an analytical solution of (3.1) as follows:

$$x(t) = x_i \cosh(t/T_c) + T_c\dot{x}_i \sinh(t/T_c) \tag{3.2}$$

$$\dot{x}(t) = (x_i/T_c)\sinh(t/T_c) + \dot{x}_i \cosh(t/T_c) \tag{3.3}$$

where $T_c = \sqrt{z_0/g}$.

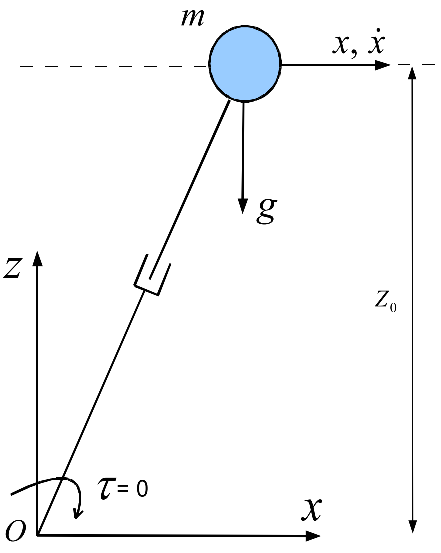


Figure 3.1: *The linear inverted pendulum model of humanoid robot. m is the total mass of robot, τ is the ankle torque*

Equations (3.2) and (3.3) are usually used to to generate nominal trajectory for the robot's body. The swing foot placement can then be planned according to the nominal body's trajectory.

Multiplying both sides of (3.1) by $\dot{x}$ and integrating it, we have

$$\frac{1}{2}\dot{x}^2 - \frac{g}{2z_0}x^2 = constant = E \tag{3.4}$$

Kajita *et al.* [39] called E the *orbital energy*. The first term of (3.4) represents kinetic energy per unit mass of the body's CG. The second term represents the potential energy caused by the force field (gravity in this case) that subjects a force of $(g/z_0)x$ on the unit mass located at x.

Fig. 3.2 shows the phase plane trajectories characterized by the value of the orbital

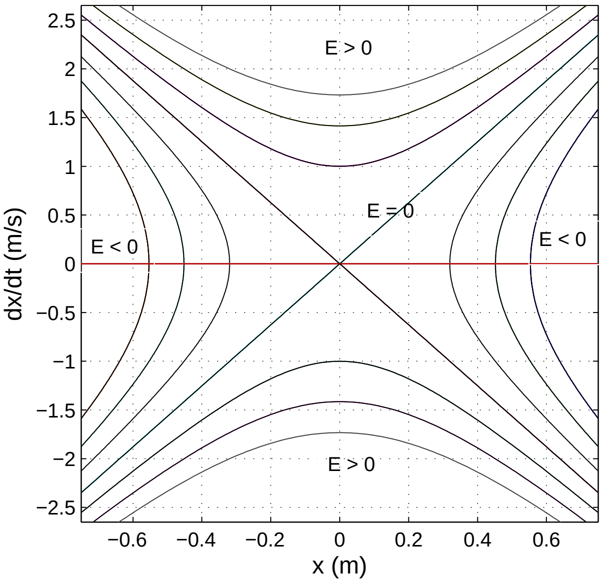


Figure 3.2: *Phase plane trajectories.*

energy. When $E > 0$, the body swings from the minus side to the plus side of the x axis or vice versa. $E = 0$ represents the equilibrium state ($x = 0, \dot{x} = 0$), the state swinging toward the equilibrium point, or the state swinging out from that point. When $E < 0$, the body never passes the point $x = 0$. In this case, the orbits have a turning point at which the velocity of the body is zero.

3.2 Gravity-compensated Inverted Pendulum Model

In the Linear Inverted Pendulum Model (LIPM), it is assumed that the legs are massless. When applying the LIPM to generate nominal trajectories for a real biped robot, there are two possible situations. If the swing leg's mass is very small compared to the total body's mass, the nominal trajectory can be easily realized. In contrast, if the swing leg's mass is not negligible compared to the total body's mass, the nominal trajectory may not be realized and the resulting walking motion may deviate from the desired walking motion.

Park *et al.*[56] improved the LIPM by introducing the term Gravity-Compensated Inverted Pendulum Mode (GCIPM). In this model, the biped robot is considered as a two-mass model: one point mass m represents the swing leg and one point mass M represents the rest of the body including the stance leg (see Fig. 3.3). Similar to the LIPM, the body mass M is assumed to move horizontally and maintain a constant height of Z_0. The ankle torque is assumed to be limp. From this model, the moment equation is derived as follows:

$$M\ddot{X}Z_0 - MgX + m\ddot{x}z - m\ddot{z}x - mgx = 0 \tag{3.5}$$

where X and x are the horizontal coordinates of the body's mass M and the swing leg's mass m, respectively; g is the gravity constant. (3.5) can be rewritten as follows:

$$\ddot{X} - \omega^2 X = F(t) \tag{3.6}$$

where $F(t) = \gamma(gx + x\ddot{z} - z\ddot{x})$, $\gamma = m/M/Z_0$ and $\omega = \sqrt{g/Z_0}$.

The function $F(t)$ represents the dynamics effect of the swing leg on the CG (Centre of Gravity) motion. If the trajectory of the swing leg is given, its influence to the whole robot's dynamics and the inverted pendulum motion of the CG can be determined.

Different choice of the foot trajectory $(x(t), z(t))$ would result in different $F(t)$ and thus

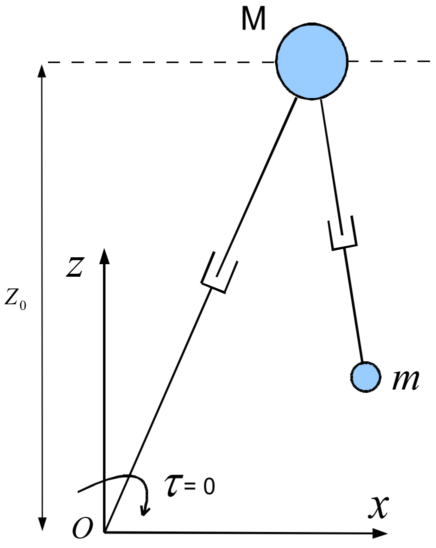


Figure 3.3: *The GCIPM of biped robot. m is the swing leg's mass, M is the total mass of the body excluding the swing leg's mass.*

different trajectory of the body's CG. In order to show a particular solution of (3.6), the swing foot trajectory is chosen as follows:

$$x(t) = -S\cos(\omega_f t) \tag{3.7}$$

$$z(t) = \frac{h_f}{2}[1 - \cos(2\omega_f t))] \tag{3.8}$$

where S is the stride length, h_f is the maximum swing foot's height, the stride frequency $\omega_f = \pi/T$, T is the one step period.

If T and h_f were chosen such that $h_f\omega_f \ll g$, the function $F(t)$ in (3.6) becomes

$$F(t) \approx -\gamma g S\cos(\omega_f t) \tag{3.9}$$

which means that the dynamics of the swing leg is dominated by its gravity term. The solution of Eq. (3.6) is then determined as

$$X(t) = C_1 e^{\omega t} + C_2 e^{-\omega t} + \eta \cos(\omega_f t) \tag{3.10}$$

with

$$C_1 = \frac{1}{2}(X(0) + 1/\omega \dot{X}_0 - \eta) \tag{3.11}$$

$$C_2 = \frac{1}{2}(X(0) - 1/\omega \dot{X}_0 - \eta) \tag{3.12}$$

$$\eta = \frac{\beta g S}{\omega^2 + \omega_f^2} \tag{3.13}$$

where variables $X(0)$ and $\dot{X}(0)$ are the initial position and the initial velocity, respectively, of the CG of the biped robot.

3.3 Effects of The Swing Leg

In this section, the dynamic effect of the swing leg on the whole robot dynamics is studied. A simple two-point-mass model similar to the above GCIPM is used (see Fig. 3.4). The point mass M represents the total mass of the whole robot except for the swing leg. The point mass m represents the mass of the swing leg. In this model, unlike the LIPM, the ankle torque is not assumed limp, which means ankle joints are equipped with actuators. Since ankle joint is available, the robot must have feet. Let us assume that the length of the feet is 20cm and the ankle joint is located in the middle point of the feet.

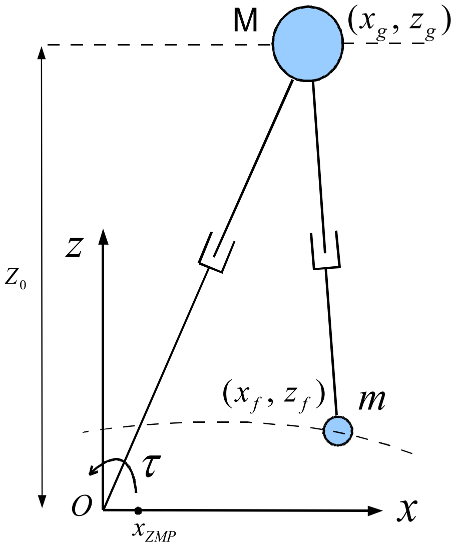


Figure 3.4: *The two-point-mass model of biped robot. m is the swing leg's mass, M is the total mass of the body excluding the swing leg's mass, x_{ZMP} is the horizontal position of the ZMP. x_g, z_g are the horizontal and vertical positions of the body's CG, respectively. x_f, z_f are the horizontal and vertical positions of the swing foot, respectively.*

To examine the effect of the swing leg's dynamics on the whole robot dynamics we implement two tests: (1) the effect of the mass of the swing leg on the zero-moment-point (ZMP) trajectory (and thus stability margin); and (2) the effect of swing foot's speed on the ZMP trajectory.

In this study, the nominal trajectory of the body's CG (the point mass *M* in this case) is generated using equations (3.2) and (3.3) of the LIPM. The nominal foot trajectory is generated using (3.7) and (3.8).

The zero-moment-point of the system in Fig. 3.4 is computed as follows:

$$x_{ZMP} = \frac{\sum_{i=1}^{N} m_i(\ddot{z}_i + g)x_i - \sum_{i=1}^{N} m_i \ddot{x}_i z_i}{\sum_{i=1}^{N} m_i(\ddot{z}_i + g)} \tag{3.14}$$

with $m_1 = M$, $m_2 = m$, $N = 2$, $x_1 = x_g$, $x_2 = x_f$, $z_1 = z_g$, $z_2 = z_f$, we have

$$x_{ZMP} = \frac{M(\ddot{z}_g + g)x_g + m(\ddot{z}_f + g)x_f - M\ddot{x}_g z_g - m\ddot{x}_f z_f}{M(\ddot{z}_g + g) + m(\ddot{z}_f + g)} \tag{3.15}$$

• **Test 1: The effect of the swing leg's mass on the robot dynamics**

In this test, different values of the swing leg's mass are used to study how the change of the swing leg's mass affect the whole robot dynamics. For no particular reason, the input parameters are chosen as follows: the body mass $M = 50$ kg, the values of the swing leg's mass used in this test are $m = 0, 3, 6, 9$ kg. The walking step length $S = 0.3$ m, the walking step time $T = 1$ s. For the sake of simplicity, the nominal trajectory is planned such that it is periodically repeatable which allows us to plan the trajectory for one walking step only. The following steps will be repeated in the same manner. With the above inputs, the nominal body's CG trajectory is obtained as shown in Fig. 3.5. The nominal swing foot trajectory generated using (3.7) and (3.8) is shown in Fig. 3.6.

Once the body's CG trajectory and foot trajectory are known, the ZMP trajectory can be computed based on (3.15). Fig. 3.7 shows the resulting ZMP trajectory for different values of m. We can see that, when $m = 0$ (the model becomes the LIPM model) the ZMP is always located at O (the ankle joint location). Since the ZMP is located at O, the ankle torque is zero. This is consistent with the LIPM model which has the assumption that the ankle torque is limp. It can also be seen from the figure that when the swing leg's mass m increases, the ZMP trajectory tends to deviate from the middle point O. The

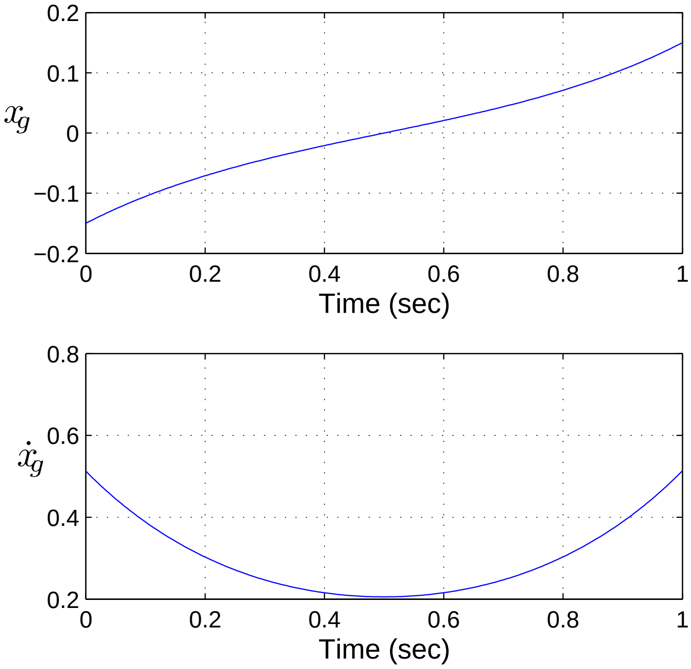


Figure 3.5: *The nominal body's CG trajectory when step length $S = 0.3$ m, step time $T = 1$ s. The upper figure shows the horizontal position trajectory while the lower one shows the horizontal velocity of the body's CG. The velocity at the start ($t = 0$) and end ($t = kT$ are equal to ensure the continuity condition of velocity when the walking motion is repeated.*

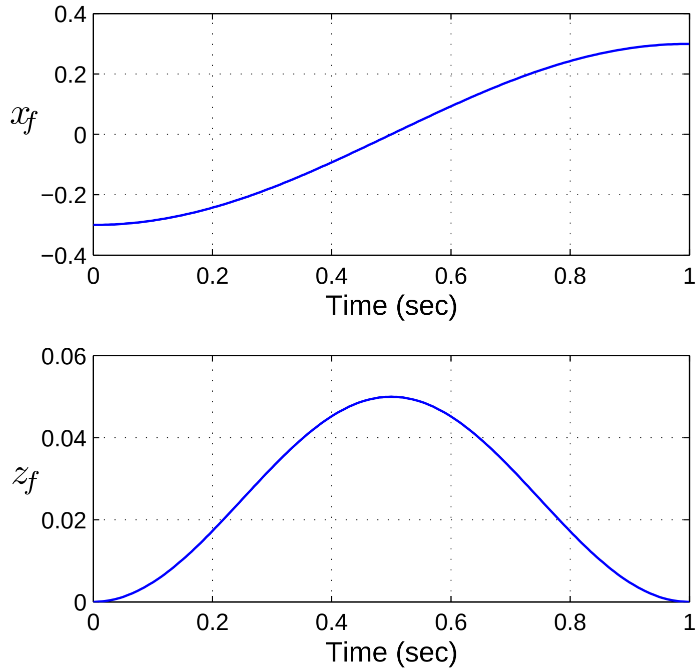


Figure 3.6: *The nominal foot trajectory with step length $S = 0.3m$ and step time $T = 1s$.*

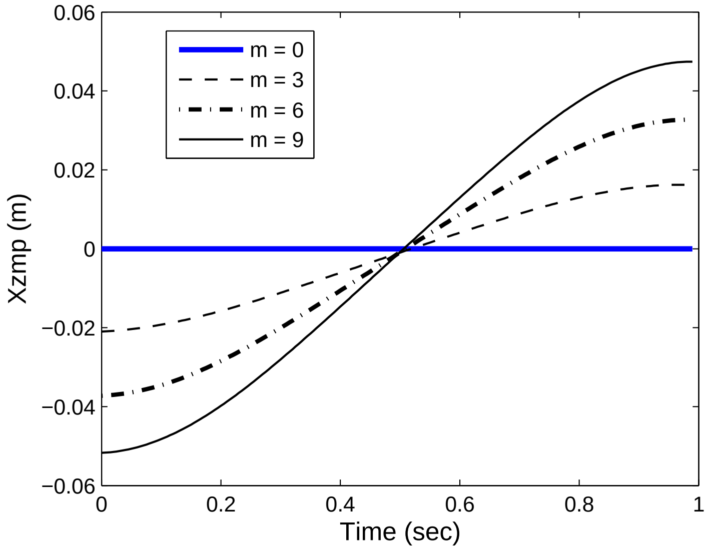


Figure 3.7: *The computed ZMP trajectories when different values of swing leg mass are used. The thick solid line shows the ZMP trajectory when $m = 0kg$, the thin dashed curve represents the ZMP when $m = 3kg$, the thick dashed-dotted curve represents the ZMP trajectory when $m = 6kg$ and the thin solid curve represents the ZMP trajectory when $m = 9kg$.*

bigger the swing leg's mass m, the more the ZMP trajectory deviates from the point O. This is not desirable because the more the ZMP is away from the point O, the smaller the stability margin. Stability margin is defined as the shortest distance from a point on the ZMP trajectory to the boundary of the stable region. In 2D case, the stable region is the distance from the heel to the toe of the supporting foot while in 3D case the stable region is the support polygon of the supporting foot/feet. When $m = 0$ kg, the stability margin is equal to 10 - 0 =10 cm (the shortest distance from foot heel/toe to the ZMP trajectory, with foot length 20cm). When $m = 3$ kg (or 6% of the body mass), the stability margin is 10 - 2 = 8 cm. When $m = 6$ kg (or 12% of the body mass), the stability margin is 10 - 3.9 = 6.1 cm. When $m = 9$ kg (or 18% of the body mass), the stability margin is 10 - 5.2 = 4.8 cm.

From the above result, it is obvious that when the swing leg mass is big enough (say more than 10% of the body mass), the dynamics of the swing leg has a significant effect on the robot's dynamics. Therefore, when m is big enough, the swing leg's dynamics should not be neglected.

- **Test 2: The effect of the swing time on the robot dynamics**

In this test, we aim to examine the effect of the swing foot's speed on the whole dynamics of the biped robot. The body mass M, the walking step length S are kept the same as in test 1. The swing leg's mass $m = 3$kg. The nominal body's CG and foot trajectories are planned using the same method as in test 1. To test the effect of swing leg's speed on the whole robot, the step time T will be altered. Fig. 3.8 shows the ZMP trajectories when the different swing times were used. It can be seen from the figure

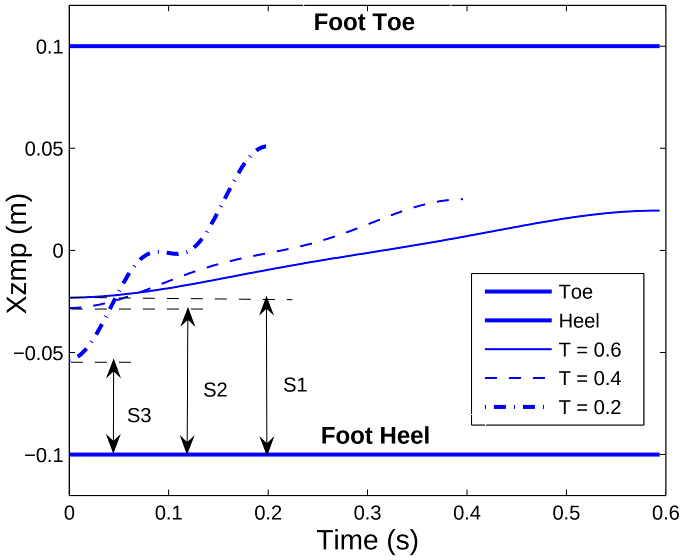


Figure 3.8: *The computed ZMP trajectories when different values of swing time were used. The continuous solid curve represents the resulting ZMP trajectory when the swing time $T = 0.6$ s. The dashed curve represents the resulting ZMP trajectory when $T = 0.4$ s and the dotted-dashed curve represents the resulting ZMP trajectory when $T = 0.2$ s. Two thick solid horizontal line represents the Foot Toe and Foot Heel, which are the stable region for 2D walking. S1, S2, S3 are the stability margins obtained when the swing time equal to 0.6s, 0.4s and 0.2s, respectively.*

that when $T = 0.6$ s, the stability margin $S1 = 7.7$ cm, when $T = 0.4$ s, the stability margin $S2 = 7.2$ cm, when $T = 0.2$ s, the stability margin $S3 = 4.7$ cm. This means that when the swing time reduced (or swing speed increased), the stability margin is reduced. When the swing speed is high enough, the swing leg dynamics become more important and should not be ignored.

3.4 Summary

The Linear Inverted Pendulum Model (LIPM) is a very useful model to analyze bipedal walking. It has compact analytical solution which allows straightforward walking gait planning. The conservation of *orbital energy* during walking is an interesting finding which can be used to support walking pattern planning. However, this model also has its own disadvantage. Since the whole robot is modeled as a point mass inverted pendulum, dynamics of independently moving parts like swing legs and arms are not considered, these un-modeled dynamics may act as a disturbing effect which deviate the robot from desired walking motion.

The Gravity-compensated Inverted Pendulum Model (GCIPM) takes into account the gravity term of the swing leg dynamics. This results in a more complete model compared to the LIPM method. According to [56], the walking gait generated using the GCIPM method is more stable than the one generated using the LIPM method. Therefore, the GCIPM method is a good improvement of the LIPM method. The GCIPM method has the advantage of a simple method whose dynamic equation can be solved analytically. However, this model is still not complete since only gravity term of the swing leg is considered. The inertial terms whose impact on robot's dynamics is significant is still not considered.

The effect of swing leg dynamics on bipedal walking was studied in this section. Two tests were used for this purpose. In the first test, the swing leg's masses were changed from small to big values to see how the swing leg's mass affects the robot's dynamics. In the second test, the swing time of the swing leg was changed to see how swinging speed affects the whole robot's dynamics. Results show that when the mass of the swing leg or the swinging speed or both are big enough, the dynamic effect of the swing leg can not be ignored.

Chapter 4

Augmented Linear Inverted Pendulum (ALIP) Model

4.1 Introduction

As discussed in Chapter 2, there are several ways to design bipedal walking patterns. Among them, the model-based approach is one of the few comprehensive and effective approaches. This approach has several advantages. Firstly, it is based on traditional dynamic analysis which is well-defined, systematic and reliable. Secondly, as long as the dynamic equation can be solved analytically, it is straightforward to design desired walking gaits. However, this approach also has drawbacks. It is usually not easy to get exact dynamic information such as the center of mass (COM) position, moment of inertia of each link of the robot. Moreover, bipedal walking is a highly nonlinear, multi-variable dynamic system whose dynamic equation can not be solved analytically most of the time unless some simplification or linearization techniques are used [72, 46, 75, 71].

The Linear Inverted Pendulum Model (LIPM) presented in Chapter 3 is a well known model-based approach using simplification technique. The dynamic equation of this model can be solved analytically without using any linearized assumptions. Although the model is very simple, it provides useful dynamic insights which are vital for designing bipedal walking gaits. When the effect of un-modeled dynamics (swing legs, arms, etc.) is negligible, the nominal trajectories generated using the LIPM method can be tracked fairly well. However, when the un-modeled dynamics is not negligible, it maybe quite challenging to control the real biped robot to follow the desired trajectory. Therefore, it is vital that the swing leg dynamics should be included in the dynamic model.

Park *et al.*[56] improved the LIPM by introducing the term Gravity-Compensated Inverted Pendulum Mode (GCIPM). Their approach takes into account the gravity of the swing leg to generate biped locomotion pattern. The walking trajectory generated by this approach is more stable than that of the LIPM. However, this method only considers the gravity term of the swing leg while the inertia terms are not considered.

In this thesis, we further improve the LIPM by proposing a new model called Augmented Linear Inverted Pendulum (ALIP). The proposed model must satisfy the following requirements: it should be as simple as possible; the dynamic equation of the model must have analytical solution; the dynamics of the swing leg must be considered.

In this model, an augmented function F is added to the dynamic equation of the Linear Inverted Pendulum. The role of the augmented function is to adjust the inverted pendulum's dynamics such that the disturbance caused by the un-modeled dynamics is minimized. The inverted pendulum's dynamics can be easily adjusted or modified by manipulating the key parameters of the augmented function. Our objective is to design a walking pattern that has the highest stability margin possible. In this study, the Zero-Moment-Point (ZMP)[83, 81] is used as a stability criterion for dynamic walking. The desired walking motion with maximized stability margin is achieved by optimizing the key parameters of the augmented function using the genetic algorithm (GA). The advantage of this method is that the disturbance caused by the un-modeled dynamics (swing legs, arms, etc.) is minimized during the optimization process. When the disturbance caused by un-modeled dynamics is minimized, better walking gait can be achieved. Detailed description of the proposed model is presented in the next section.

4.2 Augmented Linear Inverted Pendulum Model

Many researchers have been using the Linear Inverted Pendulum Model to generate nominal walking patterns[32]. Fig. 4.2 shows an example of reference hip trajectory generated using the LIPM equation (3.1) with step length $S = 0.3$ m, step time $T = 1$ s. Ideally, if there is no disturbance (caused by the un-modeled dynamics such as legs or arms, foot impact and other external forces) this reference motion can be well tracked and the zero-moment-point (ZMP) stays exactly at the ankle joint position. However, in most cases, at least one of these disturbances is present. These disturbances may cause

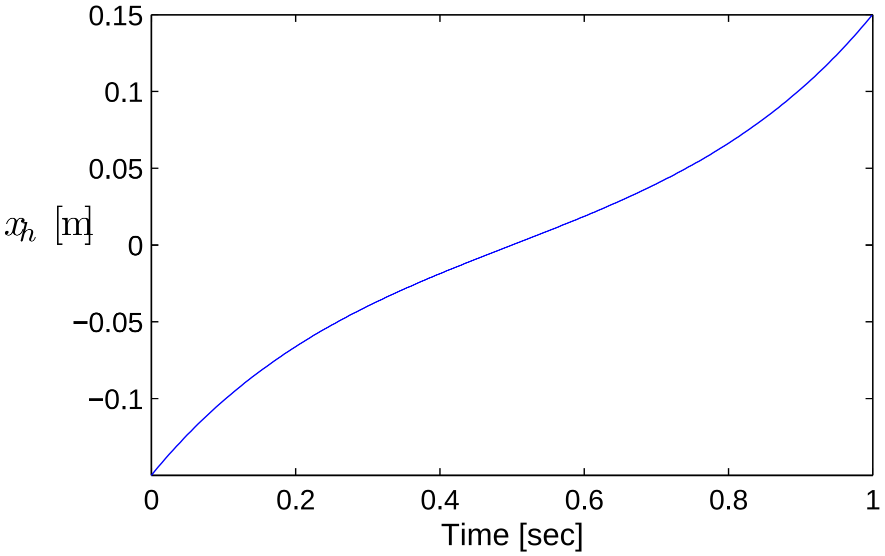


Figure 4.1: *A sample hip trajectory generated using the LIPM where step length S = 0.3 m, step time T = 1 s.*

the resulting motion to deviate from the reference motion.

In order to reduce or compensate for the disturbances caused by un-modeled dynamics, there are two possible solutions: i) use the ankle torque to force the robot to follow the reference motion[37, 38]; and ii) modify the reference motion of the center of mass (COM) so that the disturbing effect caused by un-modeled dynamics is minimized. The first solution works well when the disturbance caused by the un-modeled dynamics is small. However, when the effect of the un-modeled dynamics is big enough, the reference motion may not be realized because the ankle torque required exceeds the acceptable limit. Whereas, if we can somehow modify the COM reference motion (second solution) in such a way that it is in harmony with legs and arms' motions, a better walking motion can be achieved. By "harmony" we mean that the motions of arms and legs have very little or no disturbing effect on the reference motion.

In this study, the determination of whether one trajectory is better than the other is based on the stability margin criterion. A trajectory is better than another one if it has larger stability margin. And by this way, we define that a trajectory having larger stability margin means that the disturbing effect on the reference motion caused by arms and legs dynamics is less.

One way to modify the reference trajectory is to modify the dynamic equation (3.1). In this study, we propose to modify the dynamic equation (3.1) by adding an augmented function F to the right hand side of the equation. The dynamic equation is as follows:

$$\ddot{x} = \frac{g}{z_o}x + F \tag{4.1}$$

where the augmented function F has the following characteristics:

- *F is continuous and able to make gradual change to the dynamics of the inverted pendulum model in (4.1).*

- *F must satisfy the condition that Equation (4.1) can be solved analytically.*

- *F should be as simple as possible.*

- *The value of F can be changed by changing some key parameters.*

We call the dynamic model described by Equation (4.1) the Augmented Linear Inverted Pendulum (ALIP) model. The purpose of adding the function F to the inverted pendulum equation is to give us the possibility to change or modify the dynamics equation (3.1) so that better trajectories can be generated. The function F works as a bridge linking the un-modeled dynamics to the linear inverted pendulum dynamics. This will be explained in more details in the next section.

It can be seen that the dynamic equation (3.1) of the Linear Inverted Pendulum Model is a special case of the second-order ordinary differential equation

$$\ddot{x} + b\dot{x} + cx = 0 \tag{4.2}$$

where $a = 1$, $b = 0$, $c = -g/z_0$.

Equation (3.1) is the mathematical representation of the LIPM, a highly simplified model of bipedal walking robots. We suspect that Equation (4.2), a more general mathematical representation compared to (3.1), might be richer in representing the dynamics of bipedal walking.We propose to choose the augmented function F to be

$$F = k_p x + k_v \dot{x} \tag{4.3}$$

where k_p and k_v are the constant parameters.

Substitute (4.3) into (4.1), we have

$$\ddot{x} = \frac{g}{z_o} x + k_p x + k_v \dot{x} \tag{4.4}$$

Equation (4.4) can be re-written as follows:

$$\ddot{x} + b\dot{x} + cx = 0 \tag{4.5}$$

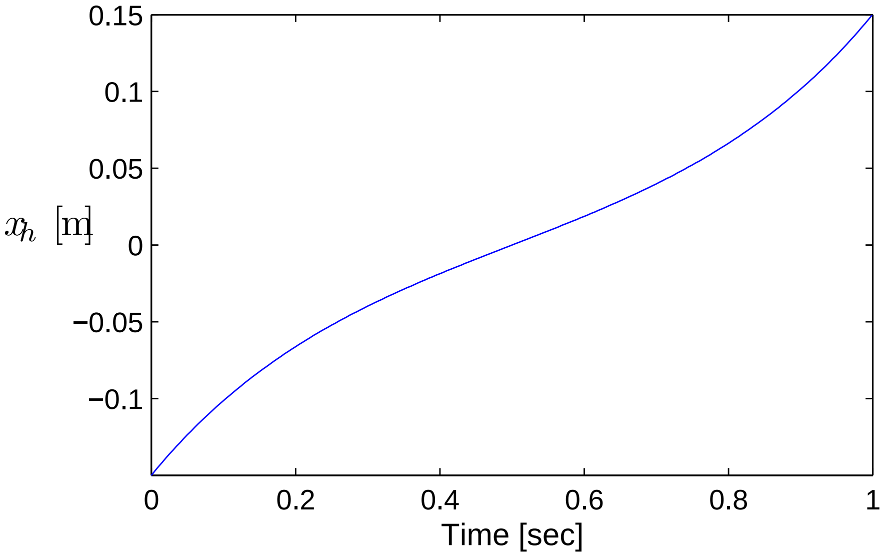


Figure 4.2: *A sample hip trajectory generated using the LIPM where step length S = 0.3 m, step time T = 1 s.*

where $b = -k_v$, $c = -kp - g/z_0$.

Solving the second-order linear differential equation (4.4), we have the following cases:

- If $b^2 - 4c > 0$:

$$x(t) = \frac{x(0)r_2 - \dot{x}(0)}{r_2 - r_1}e^{r_1 t} + \frac{\dot{x}(0) - x(0)r_1}{r_2 - r_1}e^{r_2 t} \tag{4.6}$$

where $x(0)$, $\dot{x}(0)$ are the initial horizontal position and velocity, respectively. r_1, r_2 are real roots of the auxiliary equation and are determined as below:

$$r_{1,2} = \frac{-b \pm \sqrt{b^2 - 4c}}{2} \tag{4.7}$$

$$\dot{x}(t) = \frac{x(0)r_2 - \dot{x}(0)}{r_2 - r_1}r_1 e^{r_1 t} + \frac{\dot{x}(0) - x(0)r_1}{r_2 - r_1}r_2 e^{r_2 t} \tag{4.8}$$

Since r_1, r_2 are functions of k_p and k_v, $x(t)$ is also function of k_p and k_v.

- If $b^2 - 4c = 0$:

$$x = [\dot{x}(0) - rx(0)]te^{rt} + x(0)e^{rt} \tag{4.9}$$

$$\dot{x} = [\dot{x}(0) - rx(0)](1 + tr)e^{rt} + x(0)re^{rt} \tag{4.10}$$

where $r = -b/2$.

- If $b^2 - 4c < 0$:

$$x = [x(0)\cos(\beta t) + \frac{\dot{x}(0) - x(0)\alpha}{\beta}\sin(\beta t)]e^{\alpha t} \tag{4.11}$$

$$\dot{x} = \dot{x}_0\cos(\beta t)e^{\alpha t} + \frac{\alpha\dot{x}_0 - (\alpha^2 + \beta^2)x_0}{\beta}\sin(\beta t)e^{\alpha t} \tag{4.12}$$

where $\alpha = -b/2$, $\beta = \sqrt{4c - b^2}/2$.

In summary,

$$x(t) = \begin{cases} \frac{x(0)r_2 - \dot{x}(0)}{r_2 - r_1}e^{r_1 t} + \frac{\dot{x}(0) - x(0)r_1}{r_2 - r_1}e^{r_2 t}, & \text{if } \Delta > 0 \\ [\dot{x}(0) - rx(0)]te^{rt} + x(0)e^{rt}, & \text{if } \Delta = 0 \\ [x(0)\cos(\beta t) + \frac{\dot{x}(0) - x(0)\alpha}{\beta}\sin(\beta t)]e^{\alpha t}, & \text{if } \Delta < 0 \end{cases} \tag{4.13}$$

$$\dot{x}(t) = \begin{cases} \frac{x_0 r_2 - \dot{x}_0}{r_2 - r_1} r_1 e^{r_1 t} + \frac{\dot{x}_0 - x_0 r_1}{r_2 - r_1} r_2 e^{r_2 t}, & \text{if } \Delta > 0 \\ [\dot{x}_0 - r x_0](1 + tr)e^{rt} + x_0 r e^{rt}, & \text{if } \Delta = 0 \\ \dot{x}_0 \cos(\beta t)e^{\alpha t} + \frac{\alpha \dot{x}_0 - (\alpha^2 + \beta^2)x_0}{\beta} \sin(\beta t)e^{\alpha t}, & \text{if } \Delta < 0 \end{cases} \tag{4.14}$$

where $b = -k_v$, $c = -k_p - \frac{g}{z_o}$ and $\Delta = b^2 - 4c$.

Now let's show that adding the function F in the form of (4.3) will result in better (larger stability margin) walking trajectories. The trajectories generated using equations (3.1) and (4.4) will be compared in terms of stability margin. The better trajectory would have larger stability margin.

Without loss of generality, we consider the case when $k_v = 0$ (or $F = k_p x$). Reference walking trajectories generated using the LIPM method (Eq. (3.1)) and the proposed ALIP method (Eq. (4.4) will be compared at different walking step-length S and step-time T. Once the reference trajectory is known, it is straightforward to compute the zero-moment-point (ZMP) trajectory. And from the ZMP trajectory it is easy to compute stability margin. Note that each reference trajectory generated using the ALIP method is optimized by finding the optimal value of k_p so that the stability margin is maximized.

Figure 4.3 shows the comparison of stability margin of the trajectories generated using the LIPM method and the proposed ALIP method. The comparison was made at different step-lengths (*S*) and step-times (T). Each pair of S, T corresponds to a trajectory. In this figure, each case corresponds to a value of *S*, i.e. case 1: $S = 0.1$ m; case 2: $S = 0.2$ m; case 3: $S = 0.3$ m; case 4: $S = 0.4$ m. For each case, the step-time $T = 0.2, 0.3, ..., 1$ sec. It can be seen from the figure that the trajectories generated using the proposed

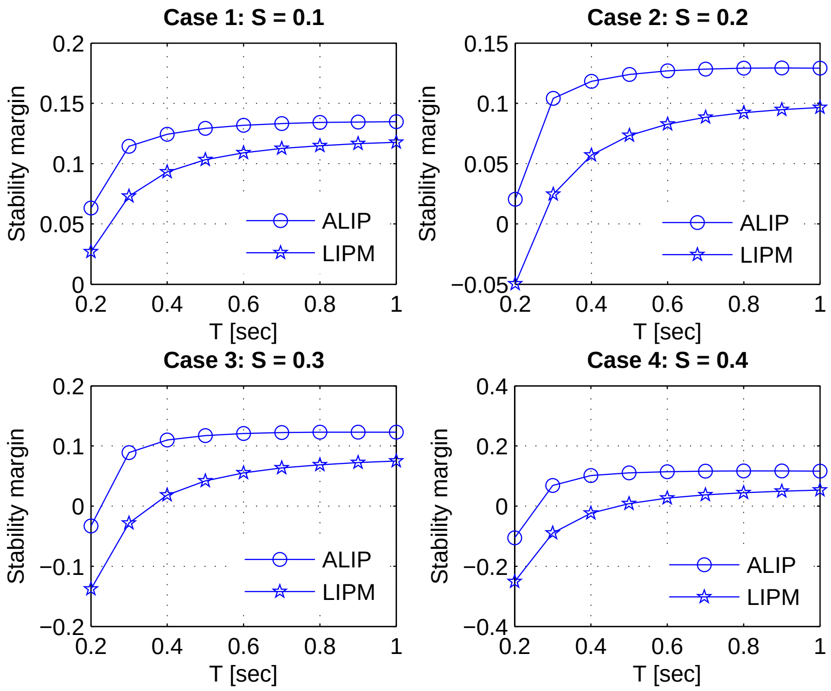


Figure 4.3: *Comparison of stability margin of the trajectories generated using the ALIP method (circle-marked curve) versus trajectories generated using the LIPM method (star-marked curve). The comparison is made at different step-lengths S and step-times T.*

method ALIP always has larger stability margin than that of the LIPM method. The corresponding optimal value of k_p and k_v for each pair of S,T is shown in Fig. 4.4.

Equations (4.13) and (4.14) will be used to plan reference trajectory for humanoid robot. Fig. 4.5 shows some sample trajectories generated using these equations when different values of k_p and k_v were used. Fig. 4.5 also shows that the function F is able to make gradual change to the LIPM dynamics.

The proposed augmented function F satisfies all the required characteristics because: 1. F is able to make gradual change to the inverted pendulum dynamics by gradually changing the parameters k_p and k_v; 2. Equation (4.1) can be solved analytically; 3. The function F is simple; 4. F can be changed by changing the key parameters k_p and k_v.

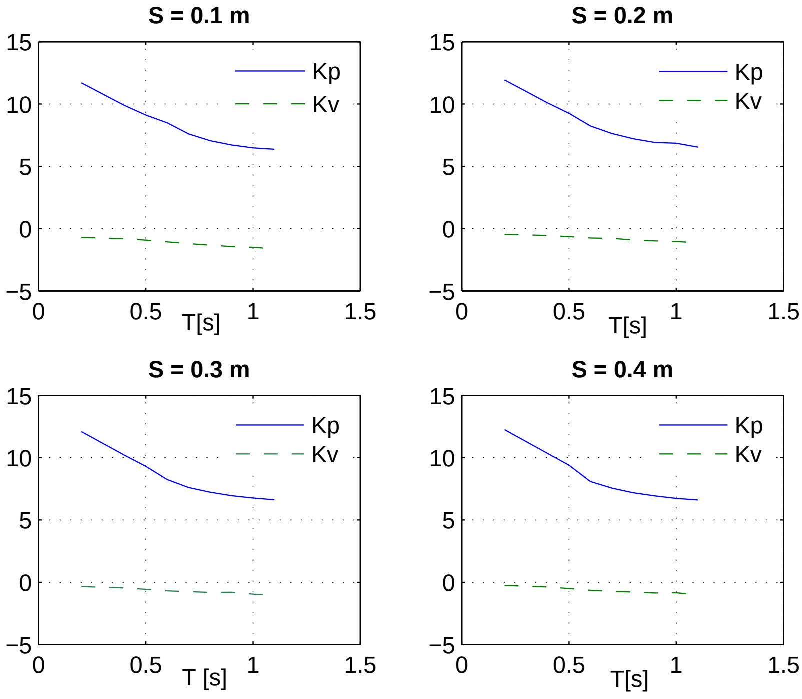


Figure 4.4: *The optimal value of k_p and k_v when* $S = 0.1$, $S = 0.2$, $S = 0.3$, $S = 0.4$

4.3 Determination of the Augmented Parameters

As mentioned earlier, the purpose of adding the augmented function F is to adjust/modify the Linear Inverted Pendulum dynamic equation so that better walking trajectories can be achieved. The modification of the dynamic equation is done by manipulating the parameters k_p, k_v of the augmented function F. The optimal value of k_p and k_v is determined using the genetic algorithm (GA) [17]. The objective of GA is to maximize the stability margin to tolerate for external disturbances during walking.

The optimal value of k_p and k_v is dependent on many factors. When the step length and step time are fixed, different robots (different height, mass distribution, etc.) would have different optimal values of k_p and k_v. Or the same robot would have different

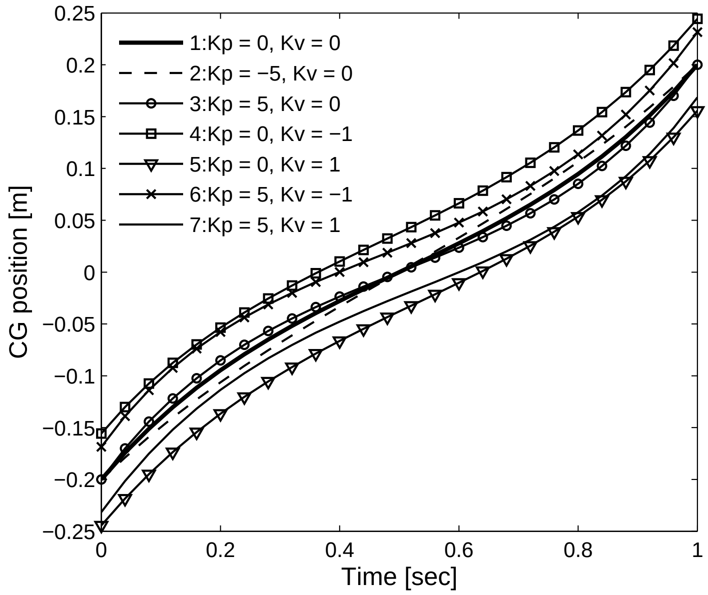


Figure 4.5: *Some sample trajectories generated using equations (4.13) and (4.14). The trajectories are numbered in sequence from 1 to 7 and each trajectory corresponds to a set value of k_p and k_v. When $k_p = 0$, $k_v = 0$ (trajectory 1 - the thick solid curve), the trajectory generated using our proposed approach will be the same as that generated using Kajita's method (LIPM). It can be seen that, the effect of k_p is to change the degree of curvature of the trajectory (see curve 2 and 3). Whereas, the effect of k_v is to offset the trajectory vertically (curve 4 and 5).*

.

sets of optimal value of k_p and k_v for different values of step length and/or step time. This is disadvantageous because whenever the step length, step time or robot's physical properties (height, mass distribution, etc.) change, the optimal value of k_p and k_v must be recomputed. However, it is understandable and acceptable because there is no such trajectory that is optimal for any robot or any step length and step time.

Although it is unrealistic to find general description of k_p and k_v for all biped robots at various walking conditions, it is possible to find general description of k_p and k_v for a specific robot operating at different values of step length and step time. This will be explained in more details in the next chapter.

Chapter 5

Off-line Walking Gait Planning in Sagittal Plane

This chapter describes the application of the proposed ALIP model to design walking gait for biped in sagittal plane. The hip trajectory is planned using the dynamic equation of the proposed ALIP model. The foot trajectory is planned using the polynomial interpolation method. The effectiveness of the proposed method is verified through two simulations. In this chapter, the genetic algorithm, an optimization tool used in finding the optimal values of the augmented parameters k_p, k_v is also presented.

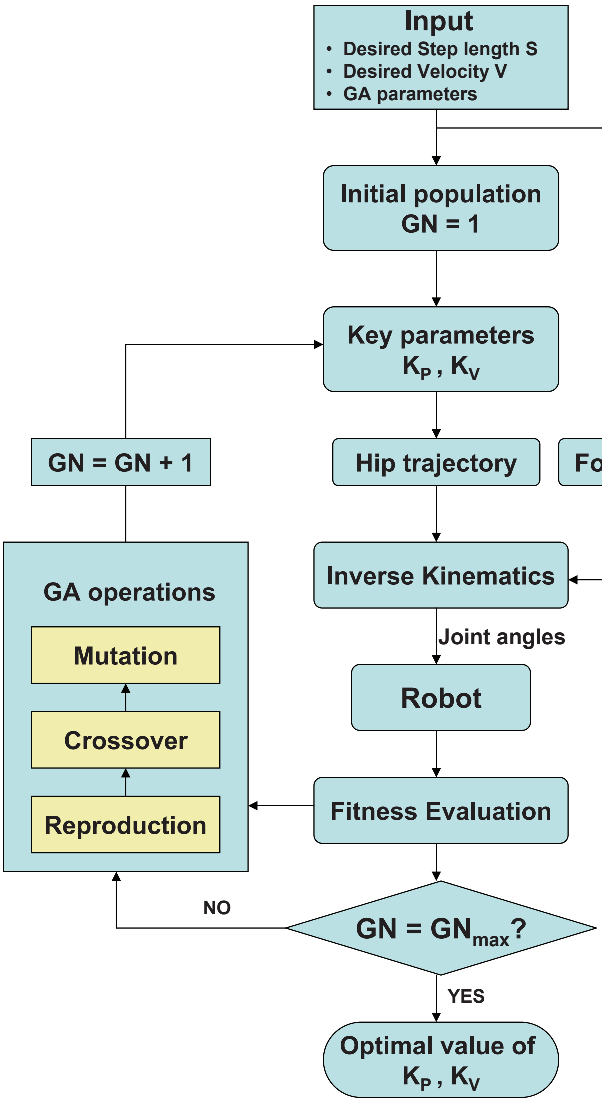


Figure 5.1: *The flowchart of the proposed algorithm. GN is the Generation Number, GN_{max} is the maximum Generation Number, k_p and k_v are the key parameters (from Equation 4.3) to be optimized*

5.1 The Proposed Algorithm

This section describes the structure of the optimization algorithm whose objective is to find the optimal values of the key parameters k_p and k_v. As shown in Fig. 5.1 the inputs to the algorithm are the desired step length S, the desired velocity V (or we can chose the desired step time T as an input), and the parameters of the genetic algorithm (GA) such as crossover rate, mutation rate, population size, maximum generation number GN_{max}. The maximum generation number GN_{max} is chosen by trial and error. GN_{max} should be large enough to make sure that GA converges after GN_{max} generations.

In the beginning of the algorithm, GA generates the initial population (or a set of initial solutions). Each individual of the population corresponds to one solution of k_p and k_v. Given the value of k_p and k_v, the hip trajectory can be computed. The foot trajectory is determined based on the desired step length and step time. Once the hip and foot trajectories are known, inverse kinematics is used to compute the reference joint angles. These joint angles information is sent to the robot's controller to control the robot to follow the desired walking pattern. Based on the position, acceleration and mass and inertia information of each part of the robot, the Zero Moment Point (ZMP) position can be computed. Once the ZMP is known, it is straightforward to evaluate the fitness function. The value of the fitness function of all the individuals are used to generate the next generation of solution. The next generation is generated by GA operations including reproduction, crossover and mutation. The process repeats until the maximum generation number GN_{max} is reached. The output of the algorithm is the optimal value of k_p and k_v.

In the following sections, the planning of hip trajectory and foot trajectory, genetic operations and the computation of the ZMP will be presented in detail.

5.2 Hip Trajectory

The hip trajectory is planned using equations (4.13) and (4.14). $x(t)$ and $\dot{x}(t)$ are functions of time and the parameters k_p and k_v. Different choice of (k_p, k_v) results in different hip trajectory. Our objective is to find the optimal value of k_p and k_v such that the resulting walking gait has highest possible stability margin. Equations (4.13) and (4.14) can be rewritten as follows:

$$x_f^{(k)} = \begin{cases} \frac{r_2 e^{r_1 T} - r_1 e^{r_2 T}}{r_2 - r_1} x_i^{(k)} + \frac{e^{r_2 T} - e^{r_1 T}}{r_2 - r_1} \dot{x}_i^{(k)}, & \text{if } \Delta > 0 \\ [e^{rT} - rTe^{rT}] x_i^{(k)} + Te^{rT} \dot{x}_i^{(k)}, & \text{if } \Delta = 0 \\ [\cos(\beta T) - \frac{\alpha}{\beta} \sin(\beta T)] e^{\alpha T} x_i^{(k)} + \frac{\sin(\beta T)}{\beta} e^{\alpha T} \dot{x}_i^{(k)}, & \text{if } \Delta < 0 \end{cases} \tag{5.1}$$

$$\dot{x}_f^{(k)} = \begin{cases} \frac{r_1 r_2 (e^{r_1 T} - e^{r_2 T})}{r_2 - r_1} x_i^{(k)} + \frac{r_2 e^{r_2 T} - r_1 e^{r_1 T}}{r_2 - r_1} \dot{x}_i^{(k)}, & \text{if } \Delta > 0 \\ -Tr^2 e^{rT} x_i^{(k)} + (1 + rT) e^{rT} \dot{x}_i^{(k)}, & \text{if } \Delta = 0 \\ -\frac{\alpha^2 + \beta^2}{\beta} \sin(\beta T) e^{\alpha t} x_i^{(k)} + [\cos(\beta T) + \frac{\alpha}{\beta} \sin(\beta T)] e^{rT} \dot{x}_i^{(k)}, & \text{if } \Delta < 0 \end{cases} \tag{5.2}$$

where $b = -k_v$, $c = -k_p - \frac{g}{z_o}$ and $\Delta = b^2 - 4c$; $x_i^{(k)}$ and $\dot{x}_i^{(k)}$ are the initial horizontal position and velocity of the COM at the start of the k^{th} step, respectively; $x_f^{(k)}$ and $\dot{x}_f^{(k)}$ are the final horizontal position and velocity at the end of the k^{th} step, respectively; $v_i^{(k)}$ and $v_f^{(k)}$ are the initial and final velocity of k^{th} step, respectively. T is the time spent for

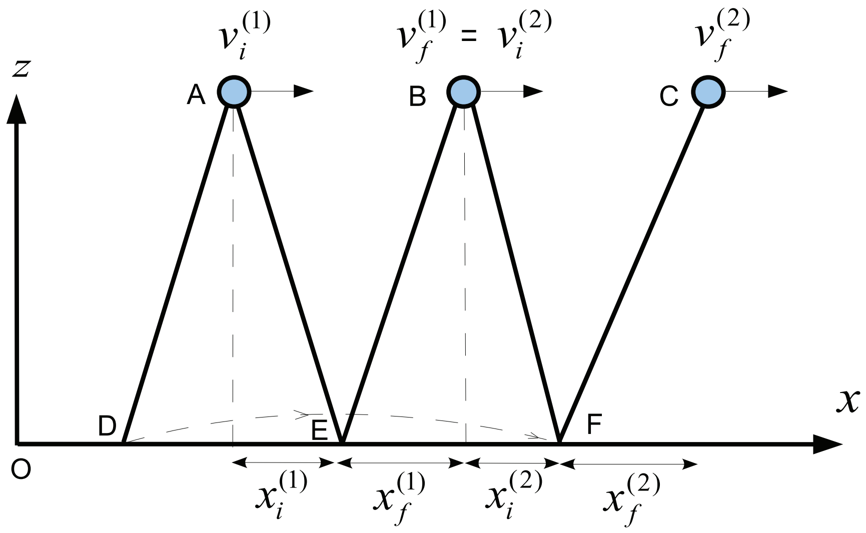


Figure 5.2: *Two consecutive steps in the sagittal plane are illustrated. In step 1, the body travels from A to B in the single-support phase (only one foot supports the robot), while the swing foot travels from D to F. The double support phase (two feet support the robot) is assumed to be instantaneous. In step 2, the body travels from B to C while the swing foot travels from E to a new point in front*

the body to travel from initial position $x_i^{(k)}$ to final position $x_f^{(k)}$.

In this thesis, to demonstrate the effectiveness of the proposed ALIP model, two types of walking gaits will be considered: i) Repetitive walking gait and ii) Non-repetitive walking gait. In the following subsections, detailed description of these two types of gaits will be presented.

5.2.1 Repetitive Walking Gait

Repetitive walking gait is the type of gait whose walking motion repeats exactly the same after each walking step. For this type of walking motion, we only need to plan the walking gait for one step, the following steps are repeated the same. The step length S

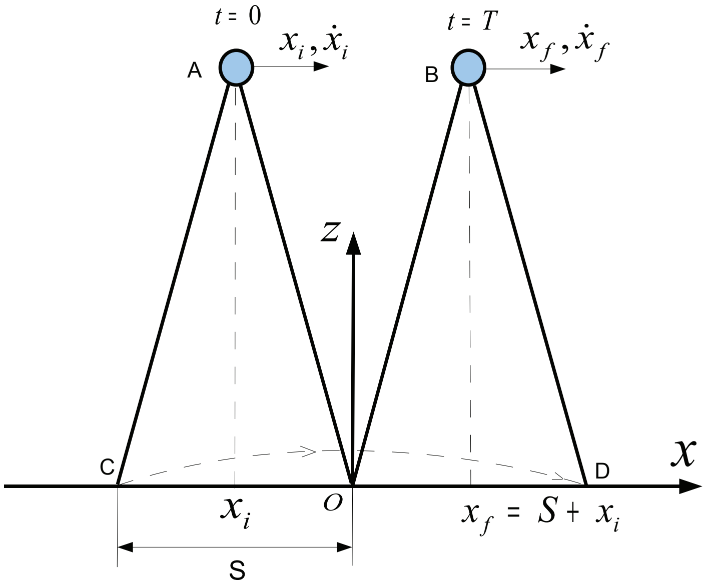


Figure 5.3: *One full step of repetitive walking gait. The body travels from A to B in the single-support phase the T seconds, while the swing foot travels from C to D. The double support phase is assumed to be instantaneous*

and step time T would be the same for every walking step. Fig. 5.3 shows one step of repetitive walking gait.

In order to have a continuous repetitive walking motion, the following continuity conditions must be imposed:

$$\begin{cases} x(0) = x_i \\ x(T) = x_f = S + x_i \\ \dot{x}(0) = \dot{x}(T) \qquad \text{(velocity continuity condition)} \end{cases} \tag{5.3}$$

Fig. 5.4 shows a sample of a repetitive walking gait. In the figure, five consecutive steps are shown with step time $T = 0.5$ s, step length $S = 0.3$ m. Note that the origin of

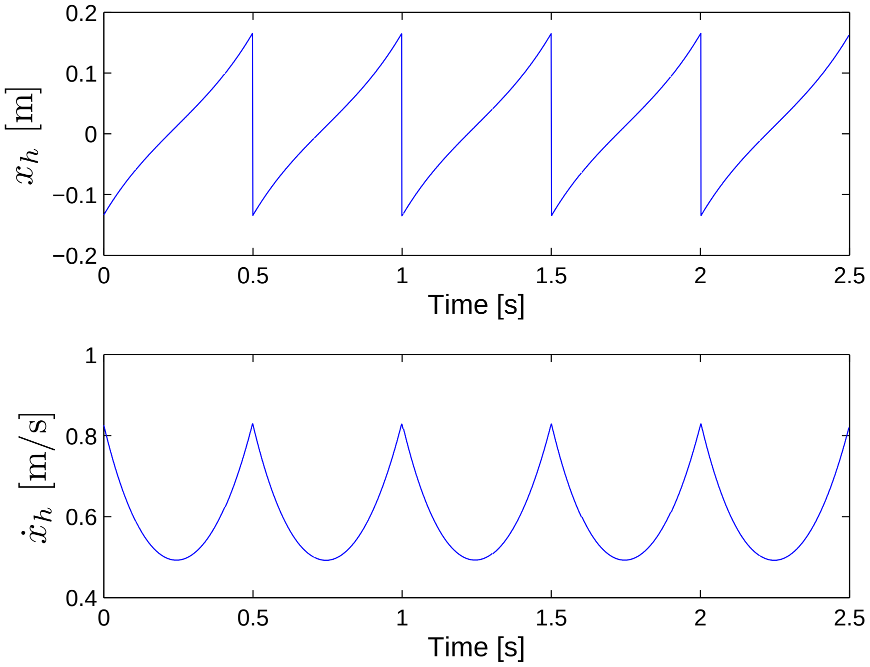


Figure 5.4: *A sample of repetitive walking gait is illustrated. Five consecutive steps are shown with step time $T = 0.5$ s, step length $S = 0.3$ m. The upper graph shows the body's COM position vs time, the lower graph shows the body's COM velocity vs time*

the coordinate frame is placed at the stance leg ankle.

5.2.2 Non-repetitive Walking Gait

In contrast to repetitive walking gait, non-repetitive walking gait does not repeat walking motion after very step. Indeed, every walking step can be different with different step length or step time. This class of walking trajectory is commonly seen when the robot is walking on uneven terrains, avoiding obstacles, walking over random stepping stones, etc. Human walking seems to be repetitive but it's actually non-repetitive because there's always differences (small or big) between each walking step. Fig. 5.5 shows a sample of non-repetitive walking gait. Five consecutive walking steps are shown. The upper

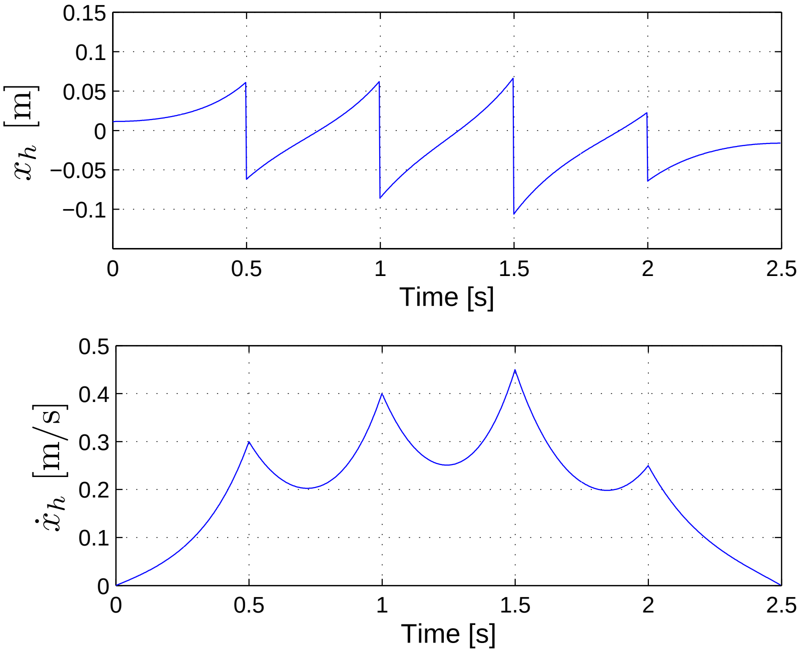


Figure 5.5: *A sample of non-repetitive walking gait is illustrated. Five consecutive steps are shown with constant step time $T = 0.5$ s and varied step length and $k_p = 10$, $k_v = 1$*

graph shows the body's COM position versus time, the lower graph shows the body's COM velocity versus time. It can be seen that the step length and velocity profile are different for each step. Note that the trajectory illustrated in Fig. 5.5 is just a sample trajectory to show what a non-repetitive trajectory looks like, not the optimal one. The optimal trajectory will be presented in the simulation results section.

5.3 Foot Trajectory

One popular method of designing the foot trajectory is to use the polynomial interpolation method. The order of the polynomial is decided based on the number of constraints. Fig. 5.6 gives an illustration of the swing foot trajectory. The swing foot starts from point A on the ground, go through the middle point B and ends at point C. Let's denote T as the total time for the foot to travel from A to C. T_1 is the time needed to travel from A to B. T_1 can be determined by assuming that it is linearly proportional to T in terms of distance traveled. The position and velocity constraints in horizontal axis are as follows:

$$\begin{cases} x_f(0) = x_A \\ x_f(T_1) = x_B = 0 \\ x_f(T) = x_C \\ \dot{x}_f(0) = 0 \\ \dot{x}_f(T) = 0 \end{cases} \tag{5.4}$$

where x_A, x_B and x_C are the given horizontal coordinates of the points A, B and C, respectively. The time T_1 can be determined as follows:

$$T_1 = \gamma T = \frac{AO}{AC}T = \frac{x_O - x_A}{x_C - x_A}T \tag{5.5}$$

Since there are five constraints, a forth-order polynomial should be enough to describe the swing foot motion. Therefore, the foot trajectory can be written in the following form:

$$x_f(t) = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \tag{5.6}$$

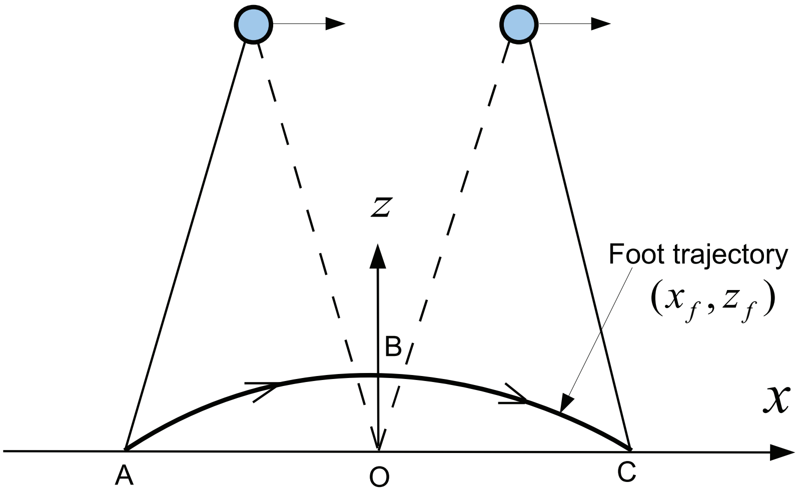


Figure 5.6: *The swing foot trajectory is illustrated as the solid curve ABC. At each step, the swing foot starts from A, a starting point on the ground, to B, a via point in the middle and finally to C, the ending point on the ground*

The coefficients computed based on the above constraints are as follows:

$$\begin{cases} a_4 = \frac{(2\gamma^3 - 3\gamma^2)(x_C - x_A) - x_A}{\gamma^4 T^4 - 2\gamma^3 T^4 + \gamma^2 T^4} \\ a_3 = \frac{2(x_A - x_C)}{T^3} - 2a_4 T \\ a_2 = -2a_4 T^2 - 1.5a_3 T \\ a_1 = 0 \\ a_0 = x_A \end{cases} \tag{5.7}$$

where $\gamma = \frac{x_O - x_A}{x_C - x_A}$.

The position and velocity constraints in vertical axis are as follows:

$$\begin{cases} z_f(0) = z_A = 0 \\ z_f(T_1) = z_B = h_f \\ z_f(T) = z_C = 0 \\ \dot{z}_f(0) = 0 \\ \dot{z}_f(T) = 0 \end{cases} \tag{5.8}$$

Similarly, there are five constraints, a forth-order polynomial is good enough to describe the foot trajectory:

$$z_f(t) = b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0 \tag{5.9}$$

The coefficients are determined as follows:

$$\begin{cases} b_4 = \frac{h_f}{\gamma^4 T^4 - 2\gamma^3 T^4 + \gamma^2 T^4} \\ b_3 = -2 b_4 T \\ b_2 = -b_4 T^2 - b_3 T \\ b_1 = 0 \\ b_0 = 0 \end{cases} \tag{5.10}$$

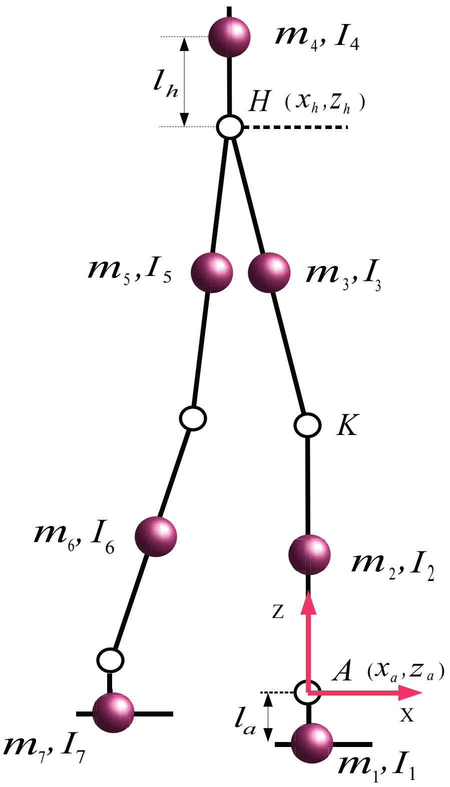


Figure 5.7: *The simple robot model showing the mass distribution of each link of the robot*

5.4 The Zero Moment Point (ZMP)

The concept of the ZMP was originally introduced by Vukobratovic in 1969. The ZMP is the location on the ground where the net moment generated from the ground reaction forces has zero components about the two axes that lie in the plane of the ground [83, 81]. The ZMP stability criterion states that the ZMP of a biped robot must be constrained within the convex hull of the foot/feet support area to ensure stability. The ZMP can be computed using the following formula:

$$x_{ZMP} = \frac{\sum_{i=1}^{7} m_i(g+\ddot{z}_i)x_i - \sum_{i=1}^{7} m_i \ddot{x}_i z_i - \sum_{i=1}^{7} I_{iy}\dot{\omega}_{iy}}{\sum_{i=1}^{7} m_i(g+\ddot{z}_i)} \tag{5.11}$$

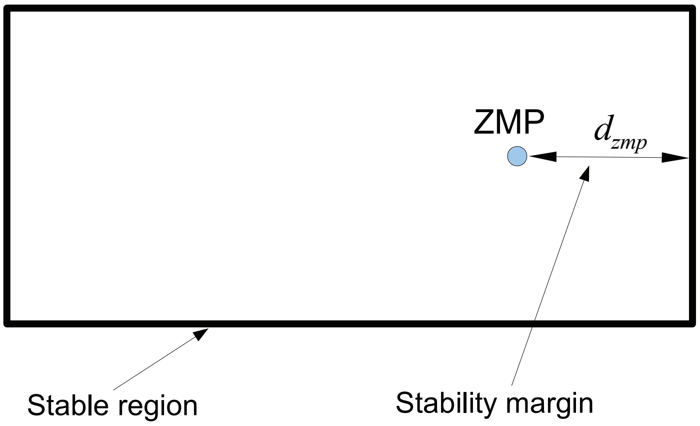


Figure 5.8: *Stable region and stability margin in sagittal plane*

$$y_{ZMP} = \frac{\sum_{i=1}^{7} m_i(g+\ddot{z}_i)y_i - \sum_{i=1}^{7} m_i\ddot{y}_i z_i + \sum_{i=1}^{7} I_{ix}\dot{\omega}_{ix}}{\sum_{i=1}^{7} m_i(g+\ddot{z}_i)} \tag{5.12}$$

where m_i is the mass of link i (Fig. 5.7). $(I_{ix}, I_{iy})^T$ is the inertia vector of link i, $(\omega_{ix}, \omega_{iy})^T$ is the angular velocity vector of link i, g is the gravitational acceleration, $(x_{zmp}, y_{zmp}, 0)$ is the coordinate of the ZMP, and (x_i, y_i, z_i) is the coordinate of the center of mass of link i on the absolute Cartesian coordinate system.

The stability margin is defined as the minimum distance from the ZMP trajectory to the boundary of the stable region (Fig. 5.8). The stability margin is maximum when the ZMP stays at the center of the stable region. In this thesis, we aim to find the walking pattern that has the largest possible stability margin.

5.5 Genetic Algorithm Implementation

5.5.1 Introduction to Genetic Algorithm

Genetic Algorithm (GA)[17] is a famous optimization algorithm developed by John Holland and his colleagues at the University of Michigan. One advantage of using GA over other algorithms is that this technique performs the search at different points simultaneously in the searching space rather than at one point. This helps to solve the problem of local optima. The other advantage of GA is that this algorithm performs the search using the fitness value as the only information. It does not require any function differentiation, so the problem of discontinuity can be avoided.

GA is usually used when dealing with complicated and highly nonlinear problems where analytical solutions are not available. Since bipedal walking is a highly nonlinear system with many degrees of freedom, quite a number of research works in this field adopted GA as an optimization tool to find optimal solution. Capi *et al.*[4], Arakawa *et al.*[2] proposed off-line trajectory planning methods for bipedal walking using GA to find the optimal key parameters of the walking gait. In these studies, minimum energy consumption is used as a criterion to guide the search. Dau *et al.*[8] applied GA to optimize the key parameters of the walking trajectory such that energy-efficient and stable walking is achieved. And many other research works [7, 29, 54, 86, 9, 10, 58, 73] used GA as an optimization tool for bipedal walking gait generation.

Initially, a population of individuals is generated randomly. GA will test each individual

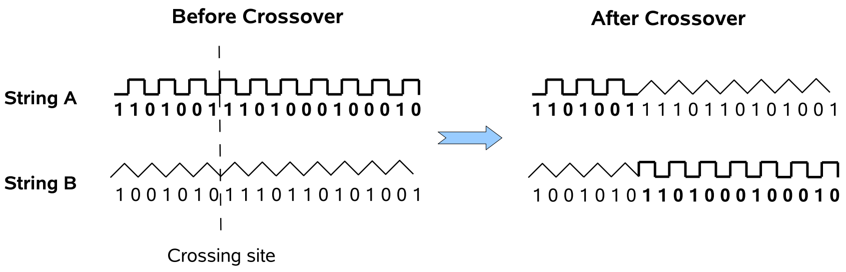


Figure 5.9: *Crossover operation*

in the population and return a corresponding fitness value. Based on these fitness values, GA's operations will be performed to generate a new generation which contains evolved individuals. These operations include reproduction, crossover and mutation.

Reproduction: During the reproduction (or selection) phase, the individuals with higher fitness values will have higher chance to be reproduced. This operation is done by putting each individual into a slot in a roulette wheel. The size of the slots is proportional to the fitness value of the individuals. After each spinning of the wheel, one individual will be selected. This operation ensures that the new generation is better than the old one.

Crossover: After the reproduction, the individuals in the new generation will be mated randomly to perform crossover operation. In this phase, each pair of randomly selected individuals will exchange their ”genes”, which is a portion of the coded string. Through this process of genes exchanging, new individuals are generated based on the existing individuals. The crossover operation can be illustrated in Fig. 5.9.

Mutation: The mutation operation occasionally modifies some specific bits in the string to try new solution. This operation sometimes brings in better individuals, but some-

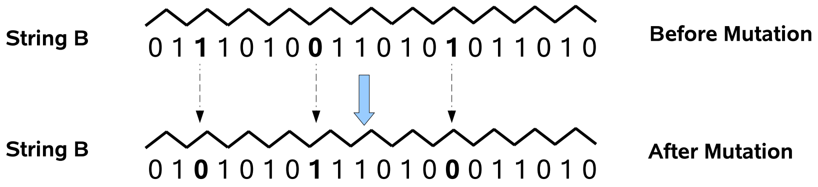


Figure 5.10: *Mutation operation*

times also destroys good individuals. So the probability for this operation to happen is usually controlled at low rates. Fig.5.10 illustrates this mutation operation.

5.5.2 GA's Variables

As mentioned in the previous chapter, the parameters k_p, k_v of the augmented function F are critical for improving the dynamic equation so that better walking trajectories can be generated. Since these two parameters are so important, their optimal value will be optimized using genetic algorithm. Therefore, the GA's variables used in this study are k_p and k_v.

5.5.3 The Fitness Function

In genetic algorithm, fitness function is the core of the searching mechanism. It has the role of guiding the search such that desired performance could be achieved. The objective that we want to achieve is reflected through the fitness value. Therefore, choosing the right fitness function is fundamentally important. Desired performance may not be obtained if the fitness function is not properly defined.

In bipedal walking, stability is probably the most critical concern of all researchers in this field. Indeed, this is true because after all the primary objective of biped robots is to move stably from one point to another without falling. Walking robots also need to be robust against external disturbances (such as being pushed, walking on uneven terrains, etc). As such, walking trajectories with highest possible stability margin is desirable. In this study, stability margin is being maximized through the fitness function. As defined earlier, stability margin is the shortest distance from the ZMP trajectory to the boundary of stable region. In sagittal plane, stability margin is the shortest distance from the ZMP trajectory to either the toe or heel of the supporting foot/feet. With the objective to maximize stability margin, the cost function is described as follows:

$$\Omega = Max\{\left|x_{zmp}^{min} - d\right|, \left|x_{zmp}^{max} - d\right|\} \tag{5.13}$$

where Ω is the cost function; d is the horizontal distance measured from the ankle joint to the middle point of the foot; x_{zmp}^{min}, x_{zmp}^{max} are the minimum and maximum ZMP position in one walking step, respectively (see Fig. 5.11). *Note that in this study the ZMP is computed based on the full dynamics of the real biped. Therefore, we can make sure that swing leg and arms dynamics are included in the trajectory planning process which helps to minimize the disturbing effects of arms and legs.* The origin O of the coordinate system is placed at the ankle joint as shown in Fig. 5.11.

In order to maximize stability margin, the cost function CF must be minimized. Since in genetic algorithm, the fitness function is always being maximized, the fitness function

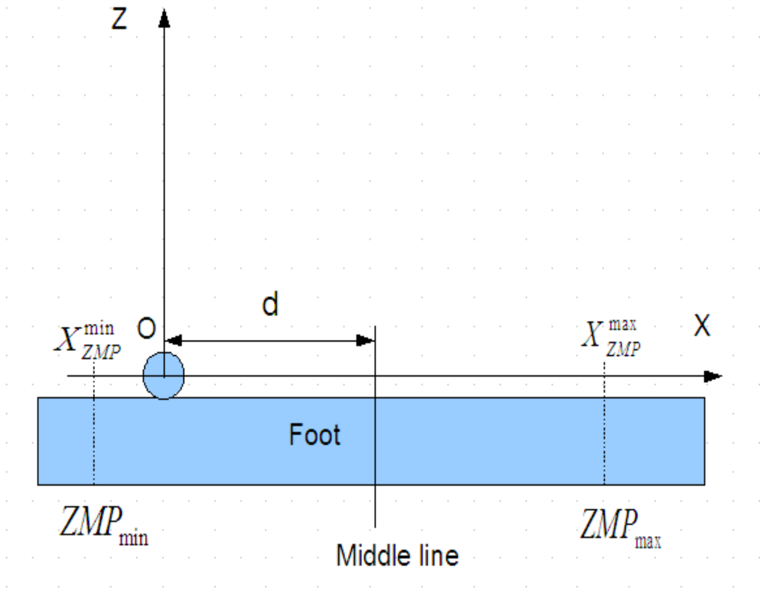


Figure 5.11: *The supporting foot is shown. The origin O of the coordinate system is placed at the ankle joint*

can be formulated as follows:

$$\Phi = \frac{1}{\Omega + \delta} = \frac{1}{Max\{\left|x_{zmp}^{min} - d\right|, \left|x_{zmp}^{max} - d\right|\} + \delta} \tag{5.14}$$

where δ is a constant positive number to ensure that the fitness function Φ is always definite even when the cost function Ω is zero.

5.6 Simulation Results

5.6.1 Repetitive Walking Motion

In this simulation experiment, repetitive walking motion is considered. The specifications of the simulated biped was taken from a biped named HUBIRO. Table 1.1 summarizes the specifications of the biped robot. The basic dimensions of the robot are shown in Figure 1.2. In this section, the walking motion is done in sagittal plane (2D walking).

For no particular reasons, the input parameters were chosen as follows: step length $S = 0.35m$, step time $T = 0.8s$. Genetic algorithm (GA) is used to determine the optimal value of k_p and k_v. The GA's parameters were chosen as follows: Number of Generations is 100, Population size is 200, Crossover rate is 0.8 and Mutation rate is 0.02.

In order for GA to start the optimization process, we need to define the ranges for the variables k_p and k_v. After some simple checks we found that when $k_p < -100$ or $k_p > 100$ the obtained trajectory is unstable (the ZMP stays outside the stable region) for any value of k_v. When $k_v < -50$ or $k_v > 50$ the obtained trajectory is always unstable for any value of k_p. Therefore, we select the ranges for k_p and k_v as follows: $-100 \leq k_p \leq 100$ and $-50 \leq k_v \leq 50$.

Since the walking motion is repetitive, the trajectory of every step will be the same. Therefore, we only need to plan the foot and hip trajectory in one step. The other steps will be repeated the same.

The optimization algorithm used to find the optimal values of k_p and k_v is shown in Fig. 5.1. GA converged after about 35 generations (see Fig. 5.12). The optimal value of the parameters k_p and k_v are $k_p = 7.591$, $k_v = -0.7346$. The computation time for the GA optimization is 665 seconds. Once k_p and k_v are determined, the hip trajectory can be computed using (5.1), (5.2) and boundary conditions (5.3). Fig. 5.13 shows the resulting

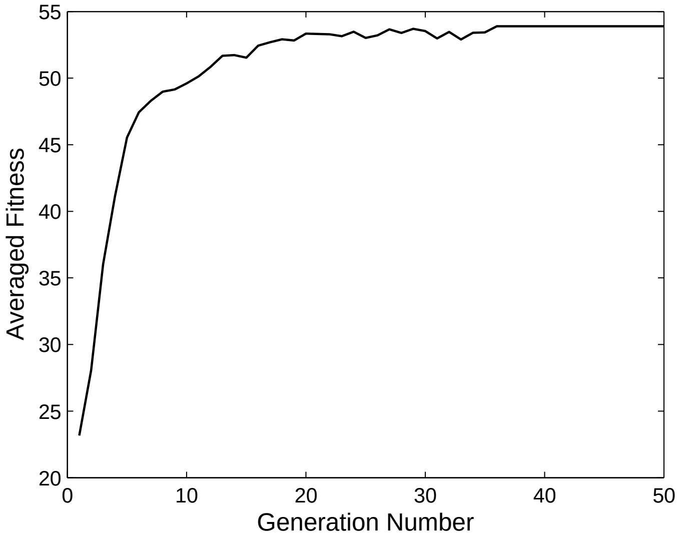


Figure 5.12: *Averaged Fitness value of each generation is shown. It can be seen from the figure that GA converged after 35 generations*

hip trajectory obtained by the proposed method ALIP. The difference between the hip trajectory obtained using ALIP model and the other two models (LIPM and GCIPM) is illustrated in Fig. 5.14.

Fig. 5.15 shows a comparison of the resulting ZMP trajectories obtained using three different methods (LIPM, GCIPM and ALIP). The thick solid continuous curve is the ZMP trajectory obtained using ALIP model while the dashed-curve is the ZMP trajectory obtained using GCIPM and the circle marked continuous curve is the ZMP trajectory obtained using the LIPM. It can be seen from the figure that the ZMP trajectory generated using the GCIPM method stays closer to the center of stable region compared to that of the LIPM method. This means that the walking trajectory generated using the GCIPM method has larger stability margin compared to the LIPM method. It can also be seen from the figure that the ZMP trajectory obtained using the proposed ALIP method stays closer to the center of the stable region compared to the ZMP generated using the

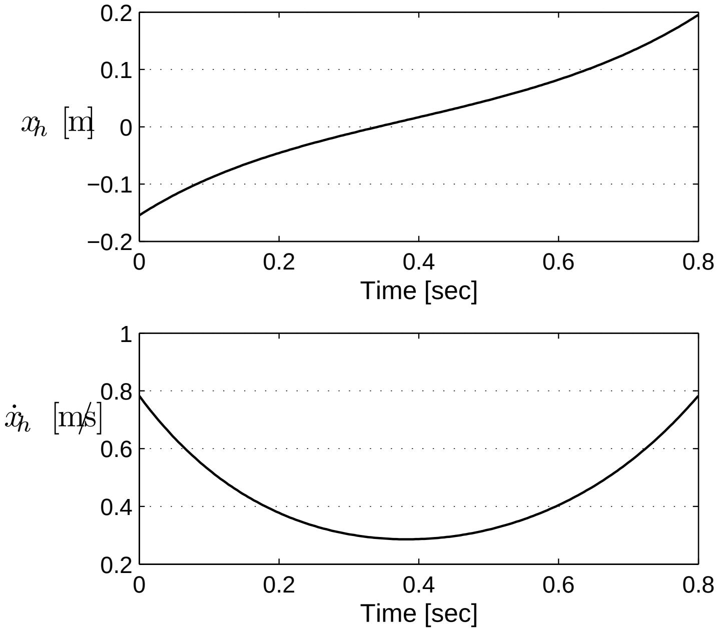


Figure 5.13: *The resulting optimal hip trajectory is shown. The upper graph shows the position trajectory x_h of the hip while the lower graph shows the velocity trajectory*

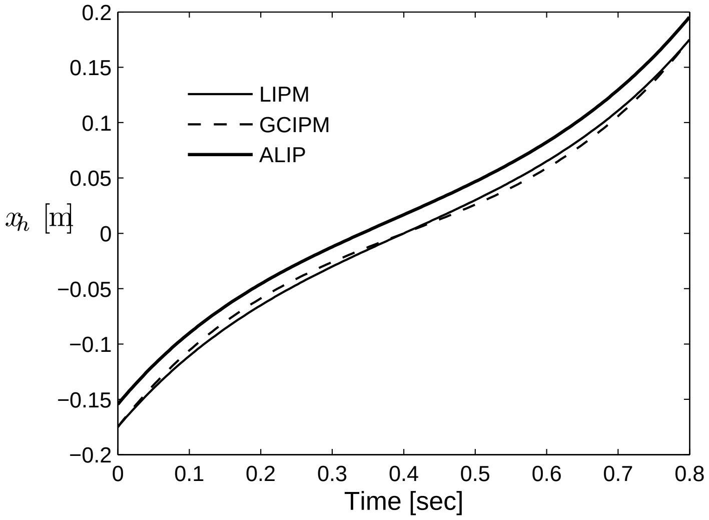


Figure 5.14: *Hip trajectories obtained using different models*

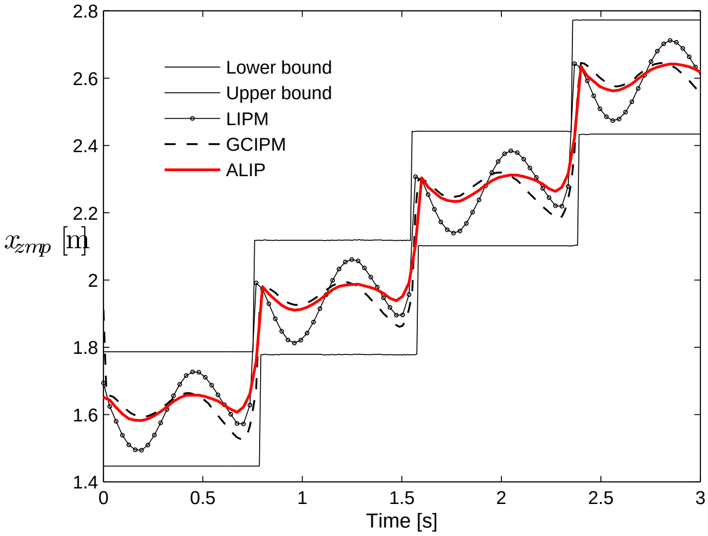


Figure 5.15: *ZMP trajectories obtained when different methods were used. The thick solid continuous curve shows the ZMP trajectory when ALIP method was used. While the circle marked continuous curve shows the resulting ZMP trajectory of the robot when the LIPM was used. And the dashed-curve shows the ZMP trajectory obtained using the GCIPM. The two thin continuous staircase-shaped curves show the stable boundaries*

GCIPM method. In conclusion, the ZMP trajectory generated using the proposed ALIP method stays closest to the center of stable region which means it has the highest stability margin. In sagittal plane, stability margin is defined as the shortest distance from a point on the ZMP trajectory to either the Heel or Toe of the supporting foot. This result shows that the walking gait generated using ALIP model is significantly more stable than that generated using the LIPM and the GCIPM. Fig. 5.16 and Fig. 5.17 show the resulting joint angle trajectories of the right leg and left leg, respectively. Fig. 5.18 shows the stick diagram of the obtained walking motion with repetitive gait.

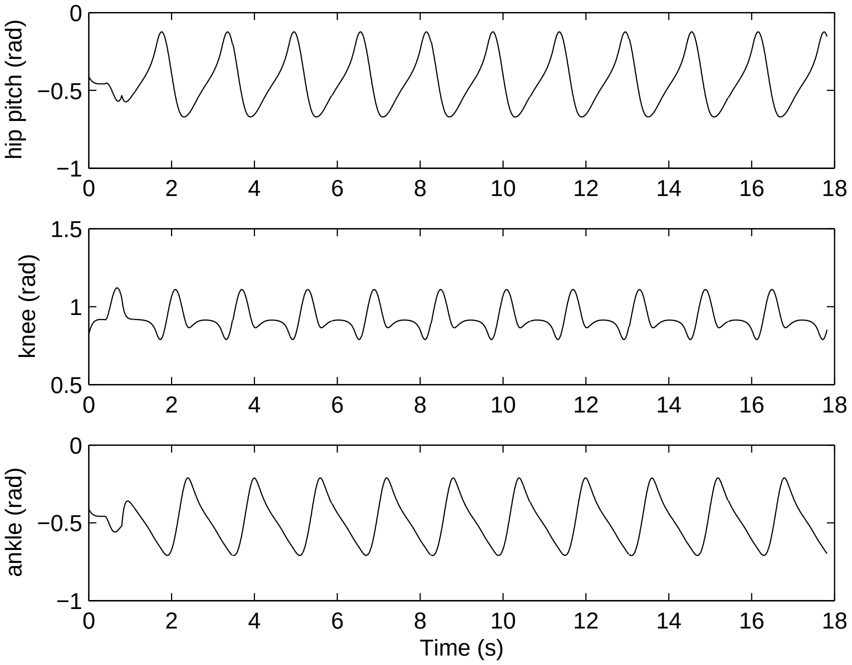


Figure 5.16: *Joint angles of the right leg*

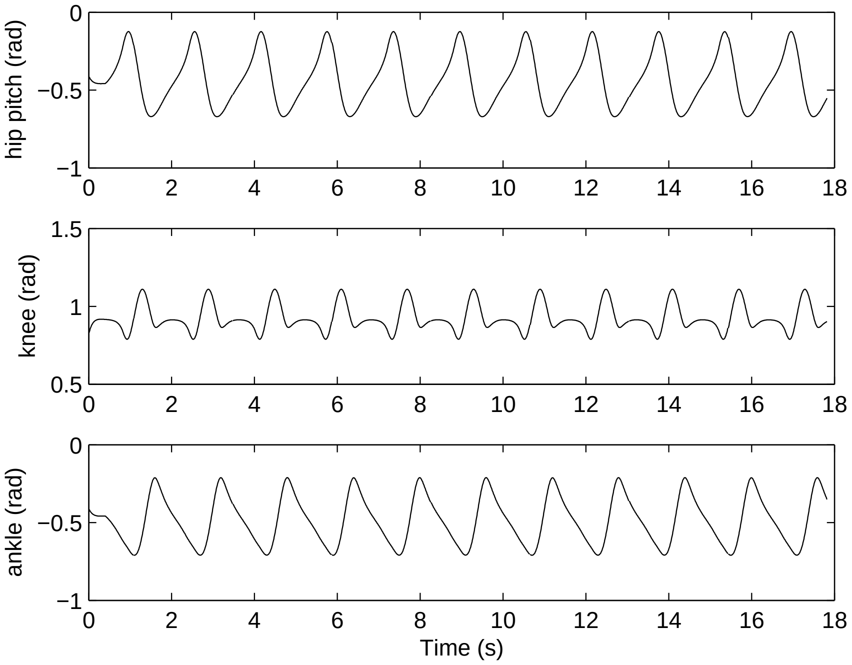


Figure 5.17: *Joint angles of the left leg*

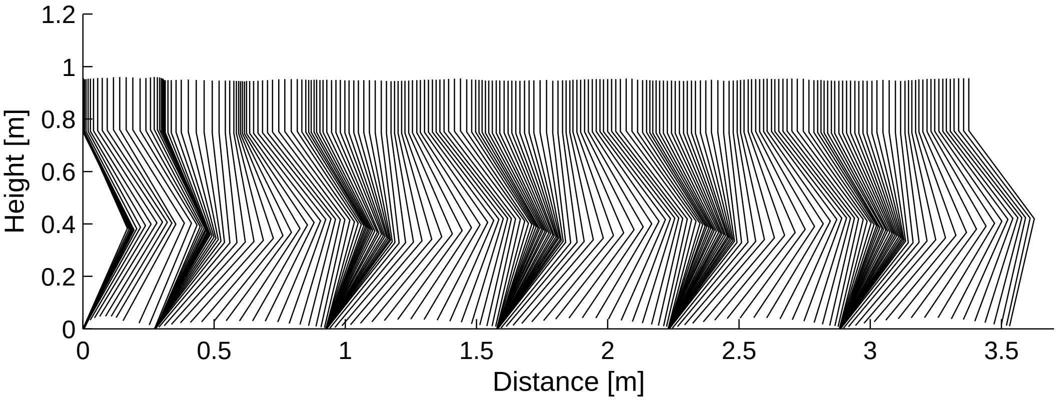


Figure 5.18: *Stick diagram of the simulated biped with repetitive walking gait. Only the right leg is shown and the image sequence is captured at 0.04 s apart.*

5.6.2 Non-repetitive Walking Motion

In this experiment, non-repetitive walking motion is considered. Basically, non-repetitive walking is a type of walking motion that is not constrained to repeat itself every step.

The non-repetitive walking gait is planned for six continuous walking steps as shown in Fig. 5.19. From the standing position A, the robot starts to walk for a few steps (the body moves from A to B, B to C, C to D, D to E, and from E to the stopping position F). The reference trajectory of the hip is planned based on the step time and the desired velocity $v_i(i = 0, 1, 2, ...6)$ at the end of step i. The step time T of each step is chosen to be $T = 0.8$ s (it is not required to choose the step time of every step to be the same. However, in this study, it is chosen to be the same for simplicity sake). The desired velocity profile of the COM is chosen as follows: $v_0 = 0$, $v_1 = 0.3$, $v_2 = 0.6$, $v_3 = 0.8$, $v_4 = 0.6$, $v_5 = 0.3$, $v_6 = 0$ (m/s).

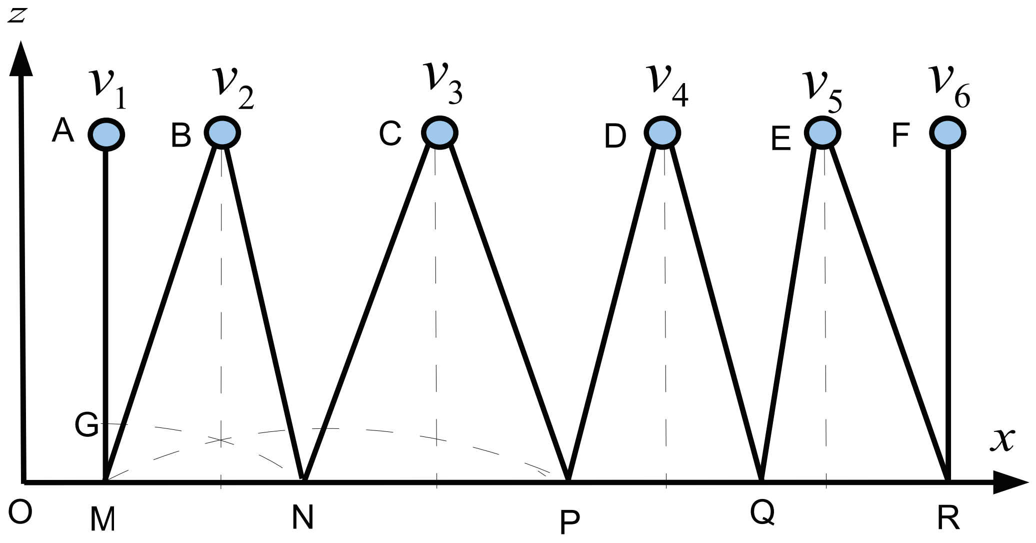


Figure 5.19: *Illustration of some steps of non-repetitive walking sequence*

Since every step the step length and velocity profile is different, the dynamics of each walking step is also different. Therefore, for each step j^{th} there should be a corresponding set of optimal value of the key parameters k_{pj}, k_{vj} $(j = 1, 2, ...6)$. Genetic algorithm was used to find the optimal value of k_{pj} and k_{vj}. The obtained optimal values for each of six walking steps are as follows: step 1: $k_{p1} = -1.719$, $k_{v1} = 2.184$; step 2: $k_{p2} = 6.516$, $k_{v2} = -0.4563$; step 3: $k_{p3} = 6.566$, $k_{v3} = -0.3916$; step 4: $k_{p4} = 5.985$, $k_{v4} = -0.51$; step 5: $k_{p5} = 5.469$, $k_{v5} = -0.6846$; step 6: $k_{p6} = 15$, $k_{v6} = 1.612$. Once the parameters k_p and k_v are known, the optimal reference hip trajectory can be easily computed using the equations 5.1 and 5.2. The optimal reference hip trajectory of non-repetitive walking motion is shown in Fig. 5.20. Fig. 5.21 shows a comparison of the ZMP trajectories generated using three different approaches (LIPM, GCIPM and ALIP). The thick solid curve shows the ZMP trajectory generated using the proposed ALIP method. The

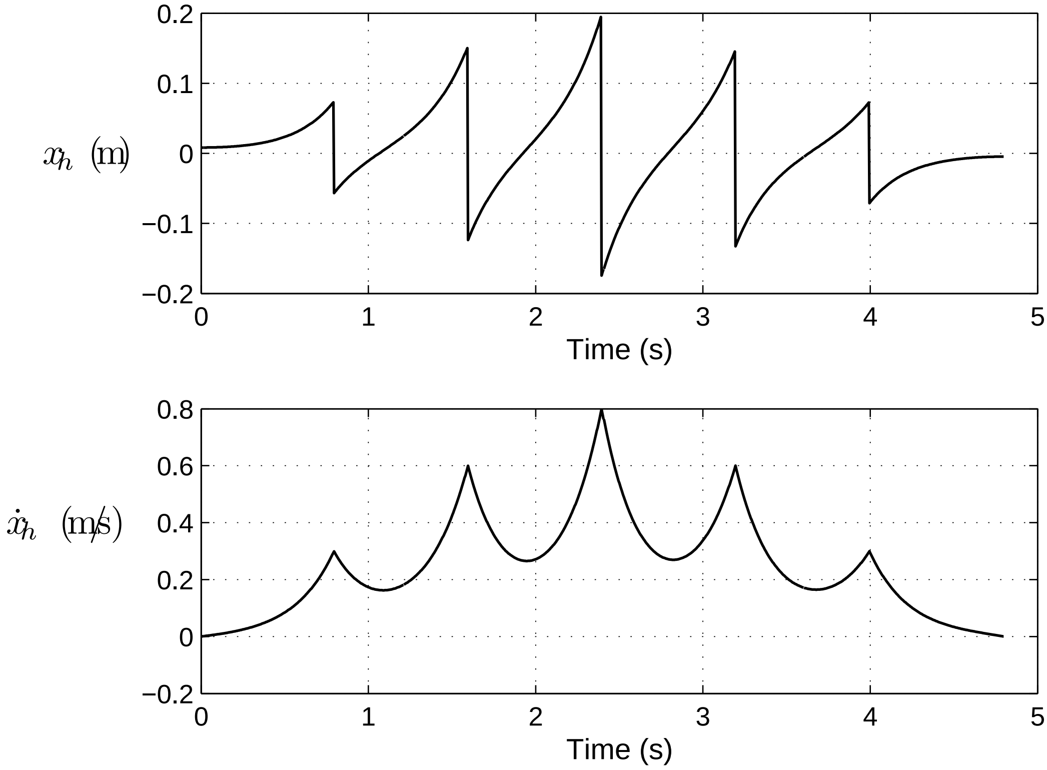


Figure 5.20: *The obtained optimal hip trajectory of six steps non-repetitive walking motion. The upper graph shows the hip position trajectory and the lower graph shows the hip velocity profile.*

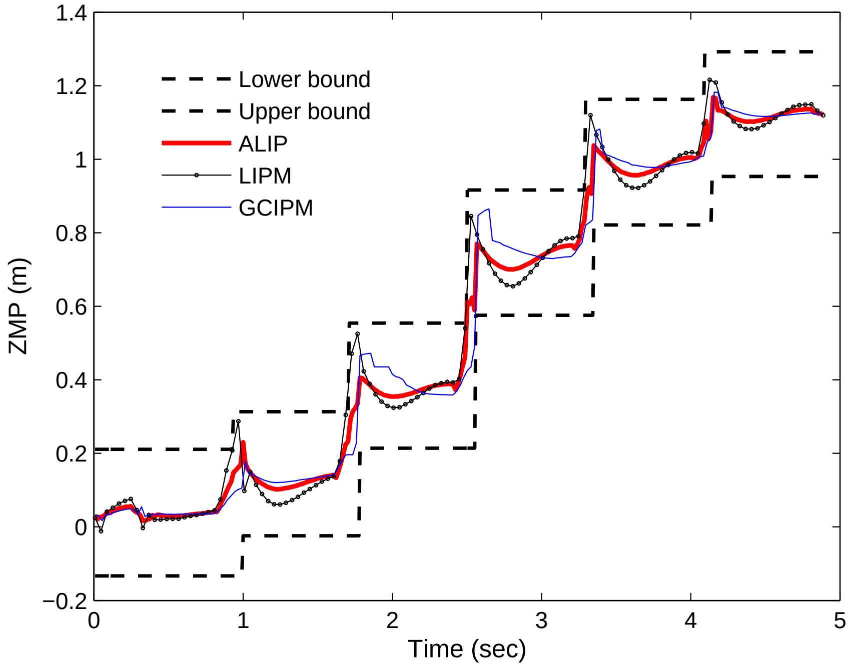


Figure 5.21: *Comparison of the resulting ZMP trajectories generated using three methods LIPM, GCIPM and ALIP. The thick solid curve shows the ZMP trajectory generated using the proposed ALIP method. The thin solid curve and the dotted continuous curve show the ZMP trajectories generated using the GCIPM method and the LIPM method, respectively.*

thin solid curve and the dotted continuous curve show the ZMP trajectories generated using the GCIPM method and the LIPM method, respectively. It can be seen from the figure that the ZMP trajectory generated using the GCIPM method stays closer to the center of the stable region. This means that the walking gait generated using the GCIPM method has higher stability margin compared to that generated using the LIPM method. It can also be seen from the figure that the ZMP trajectory generated using the proposed ALIP method stays much closer to the center of the stable region compared to the ZMP trajectory generated using the GCIPM method. Therefore, it can be concluded that the walking gait generated using the proposed method has the highest stability margin among the three mentioned methods. Fig. 5.22 and 5.23 show the resulting joint angle

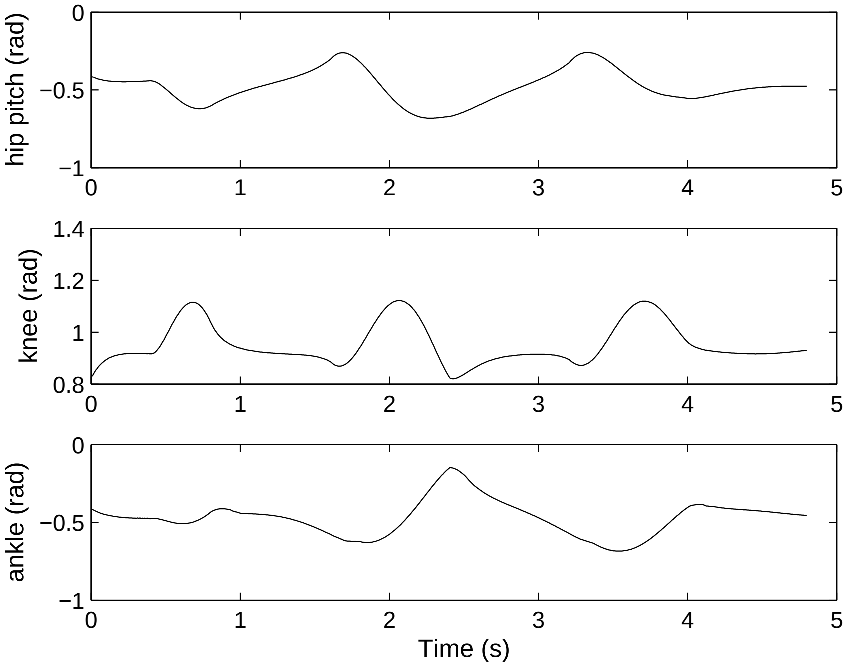


Figure 5.22: *From top to bottom are the joint trajectories of the hip, knee and ankle joints of the RIGHT leg.*

trajectories of the right and left legs, respectively. Fig. 5.24 shows the stick diagram of the simulated walking motion captured at 0.04 s apart.

5.6.3 Increase Stability Using Ankle Pitch Strategy

Biomechanics studies have shown that human beings use the ankle joint as one of the strategies to maintain balance during walking. In their study, Horak and Nashner[23] discovered that human subjects in their experiments utilized the ankle joint to counter the external disturbance and maintain balance. Inspired by this finding, in this study we aim to explore the ankle control strategy to improve walking stability.

For a stable bipedal walking motion, the zero-moment point (ZMP) and the Center of

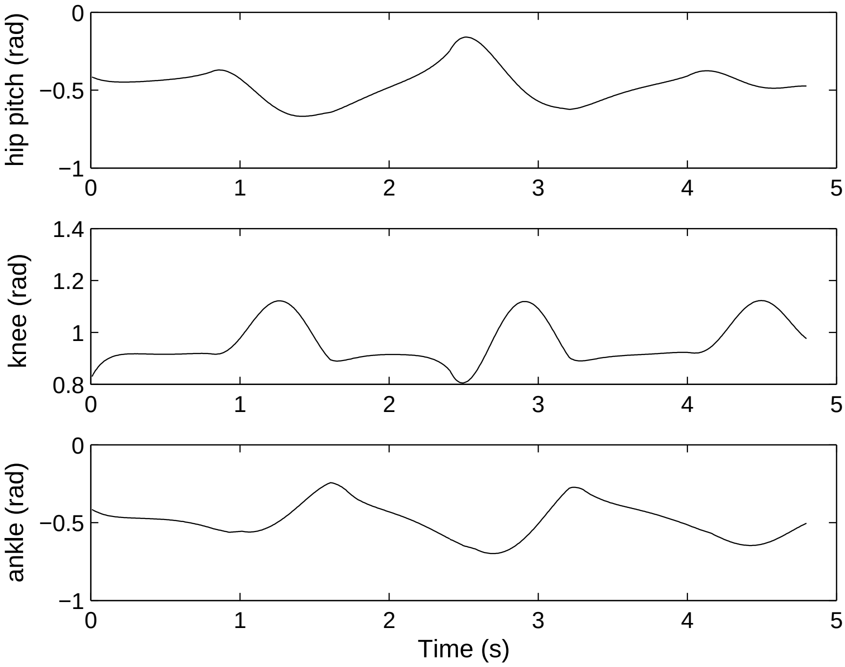


Figure 5.23: *From top to bottom are the joint trajectories of the hip, knee and ankle joints of the LEFT leg.*

Pressure (COP) are the same. In this case, the ZMP (or COP) can be computed based on the ground reaction force on the walking robot. The simulated biped has four force sensors attached at the bottom of each foot (two at the toe and two at the heel) to measure the ground reaction force. In sagittal plane, the reaction force are as shown in Fig. 5.25. If the reference coordinate system is placed as in Fig. 5.25, the location of the COP (or ZMP) can be determined as follows:

$$x_{ZMP} = x_{COP} = \frac{F2 * L_f}{F1 + F2} \tag{5.15}$$

where x_{ZMP}, x_{COP} are the horizontal location of the ZMP and COP, respectively; $F1$ and $F2$ are the total ground reaction force at the heel and the toe, respectively; L_f is the length of the foot.

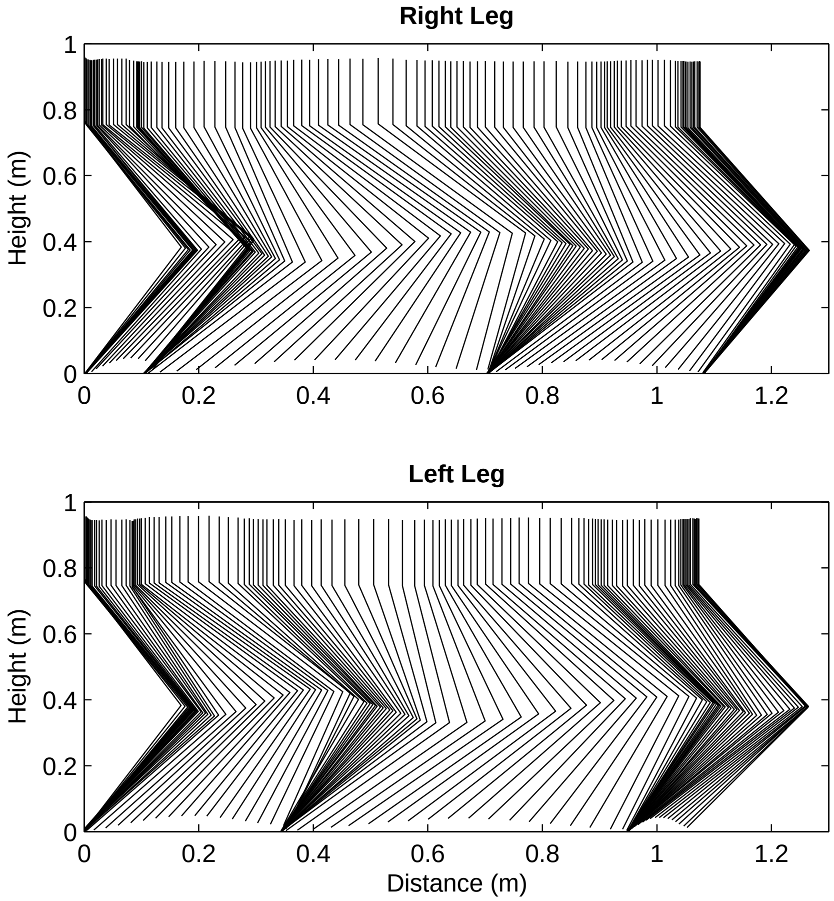


Figure 5.24: *The stick diagram of the non-repetitive walking motion simulation. The images are captured at 0.04 s apart.*

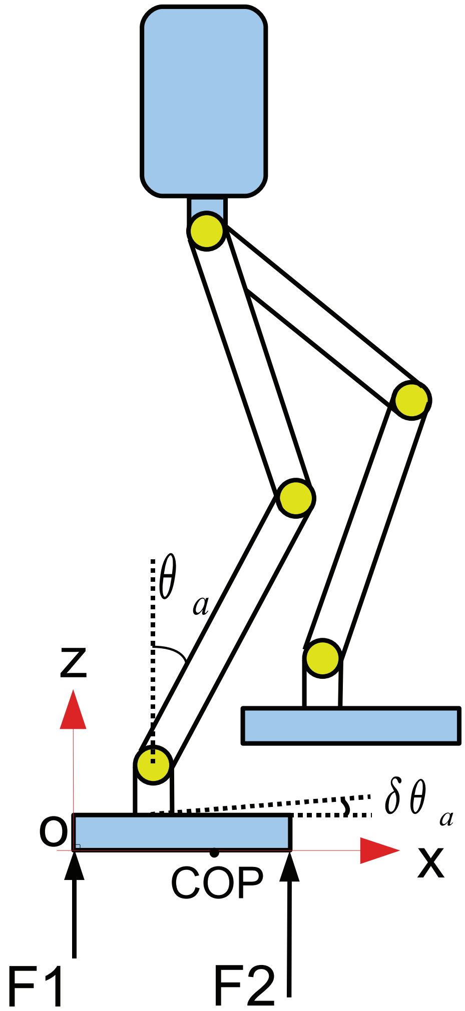


Figure 5.25: *Ground reaction force acting on the robot. $F1$ is the total reaction force at the Heel, $F2$ is the total reaction force at the Toe, COP is the location of the Center of Pressure.*

It is desirable that the ZMP stays as close to the middle point of the foot as possible (the closer the ZMP to the middle point of the foot, the higher the stability margin). Based on the equation 5.15, the ZMP is at the middle point only when $F1 = F2$. Therefore, to control the ZMP position, one could manipulate the magnitude of the reaction forces $F1$, $F2$ by controlling the angle position of the ankle joint. The controller is proposed as follows:

$$\theta_a = \theta_a^{ref} + \delta\theta_a = \theta_a^{ref} + K_a(F1 - F2) \tag{5.16}$$

where θ_a is the ankle joint angle to be controlled, θ_a^{ref} is the reference ankle joint angle computed based on the reference trajectory, $\delta\theta_a$ is the compensation amount to be added to the ankle joint, K_a is the ankle gain.

The controller in (5.16) is applied to the same repetitive walking simulation presented in section 5.6.1 to check the effectiveness of the proposed controller. Fig. 5.26 shows a comparison of ZMP trajectories obtained with and without applying the ankle control strategy. The thin continuous curve shows the resulting ZMP trajectory obtained without using ankle compensation strategy while the thick continuous curve shows the ZMP trajectory obtained when the ankle compensation strategy is applied. It can be seen from the figure that in the case when the ankle compensation strategy is adopted the ZMP trajectory obtained stays closer to the middle of the stability region (the area formed by the upper bound and the lower bound). This means that the proposed ankle control strategy helps improve the stability margin.

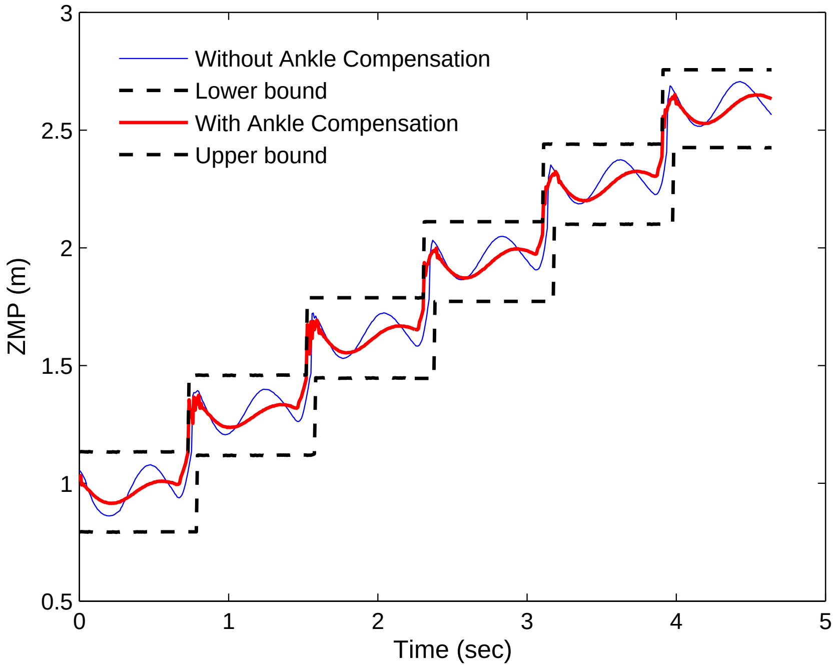


Figure 5.26: *Comparison of ZMP trajectory between the two cases: With and without using ankle compensation strategy described in (5.16). The thin continuous curve shows the resulting ZMP trajectory obtained without using ankle compensation strategy while the thick continuous curve shows the ZMP trajectory obtained when the ankle compensation strategy is applied.*

Chapter 6

Gait Planning in Frontal Plane and 3D Walking Simulation

This chapter describes the implementation of the proposed model in the frontal plane. It is assumed that the motion in the three orthogonal planes can be independently treated. The motion planned in the frontal plane is combined with the sagittal motion to create 3D dynamic walking. In this chapter, the Ankle Roll Control Strategy is used to improve stability margin in the frontal plane.

6.1 Frontal Plane Motion Planning

The 3D motion of the walking robot is divided into three orthogonal planes (sagittal plane, frontal plane and transverse plane). It is assumed in this study that the motion in these three planes are independent. The motion planning on the sagittal plane was already presented in chapter 5 hence it will not be repeated here. In the transverse plane, the main concern is the body orientation in the yaw direction. Since in this thesis, we only consider straight forward walking motion, the control task in the transverse plane becomes a simple task of controlling the desired yaw angle of the body to be zero. In this section, the focus is on the planning of the walking pattern in the frontal plane. Equations (5.1) and (5.2) are used to plan the hip trajectory in the frontal plane. These equations are rewritten below with the variable x replaced by the variable y.

$$y(t) = \begin{cases} \frac{r_2 e^{r_1 t} - r_1 e^{r_2 t}}{r_2 - r_1} y_0 + \frac{e^{r_2 t} - e^{r_1 t}}{r_2 - r_1} \dot{y}_0, & \text{if } \Delta > 0 \\ [e^{rt} - rte^{rt}] y_0 + te^{rt} \dot{y}_0, & \text{if } \Delta = 0 \\ [\cos(\beta t) - \frac{\alpha}{\beta} \sin(\beta t)] e^{\alpha t} y_0 + \frac{\sin(\beta t)}{\beta} e^{\alpha t} \dot{y}_0, & \text{if } \Delta < 0 \end{cases} \tag{6.1}$$

$$\dot{y}(t) = \begin{cases} \frac{r_1 r_2 (e^{r_1 t} - e^{r_2 t})}{r_2 - r_1} y_0 + \frac{r_2 e^{r_2 t} - r_1 e^{r_1 t}}{r_2 - r_1} \dot{y}_0, & \text{if } \Delta > 0 \\ -tr^2 e^{rt} y_0 + (1 + rt) e^{rt} \dot{y}_0, & \text{if } \Delta = 0 \\ -\frac{\alpha^2 + \beta^2}{\beta} \sin(\beta t) e^{\alpha t} y_0 + [\cos(\beta t) + \frac{\alpha}{\beta} \sin(\beta t)] e^{rt} \dot{y}_0, & \text{if } \Delta < 0 \end{cases} \tag{6.2}$$

where $b = -k_v$, $c = -k_p - \frac{g}{z_o}$ and $\Delta = b^2 - 4c$, $\alpha = -b/2$, $\beta = \sqrt{4c - b^2}/2$, $r_{1,2} = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$; y_0 and $\dot{y}_0$ are the initial horizontal position and velocity of the COM at the time t_0, respectively; $y(t)$ and $\dot{y}(t)$ are the position and velocity at the time t, respectively. Figure 6.1 shows the motion of the robot's COM in the frontal plane in one walking step. The origin O of the reference coordinate system Oxyz is placed at the ankle joint of the supporting leg. This means Oxyz is placed at the right ankle when the right leg is the

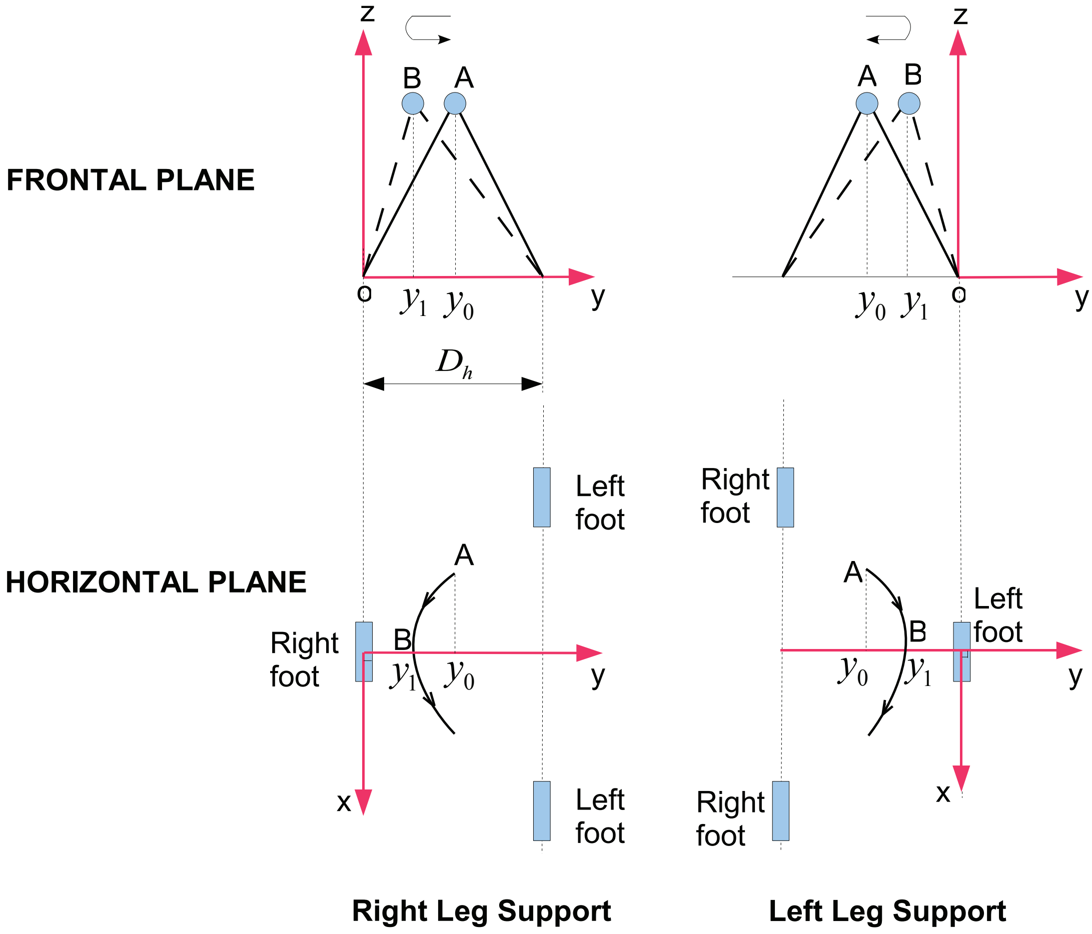


Figure 6.1: *Projections of the motion of the inverted pendulum on frontal and horizontal planes*

supporting leg and at the left ankle when the left leg is the supporting leg. When two legs are supporting legs, the coordinate system is placed at the ankle joint of the trailing leg.

In frontal plane, the stable region is defined by the lower bound and upper bound as shown in Fig. 6.2. In single support phase, the lower bound is the outer edge of the supporting foot (Fig. 6.2), the upper bound is the inner edge of the supporting foot. In

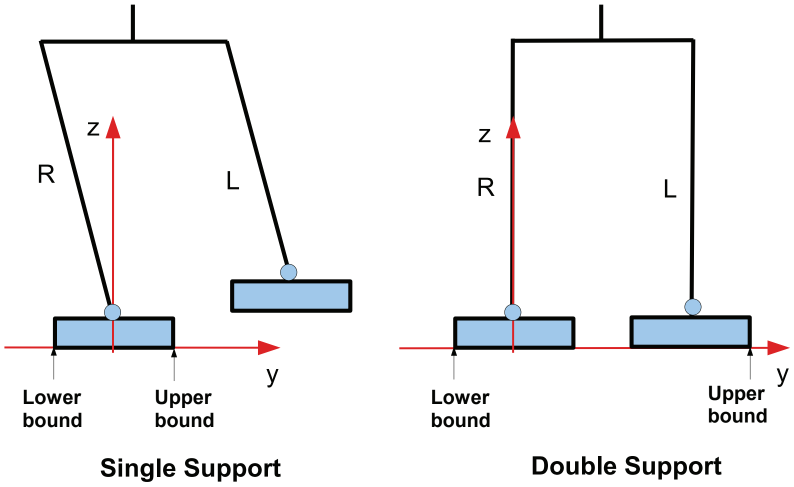


Figure 6.2: *Definition of stable region in the frontal plane*

double support phase, the lower bound is the outer edge of the right foot while the upper bound is the outer edge of the left foot.

To obtain the optimal hip trajectory in frontal plane, the key parameters k_p and k_v need to be optimized. The optimization algorithm for finding the optimal k_p and k_v is similar to that used in the sagittal plane. In this 3D simulation experiment, we will use the same inputs as in the repetitive walking case presented in chapter 5. For reader's convenience, the inputs are repeated here as follows: Step length $S = 0.35m$, Step time (the time needed to take one step) $T = 0.8s$. Genetic algorithm (GA)is used to determine the optimal value of k_p and k_v. The GA's parameters were chosen as follows: Number of Generations is 100, Population size is 200, Crossover rate is 0.8 and Mutation rate is 0.02.

The optimization algorithm used to find the optimal values of k_p and k_v is shown in Fig. 5.1. GA converged after about 40 generations. The optimal value of the parameters

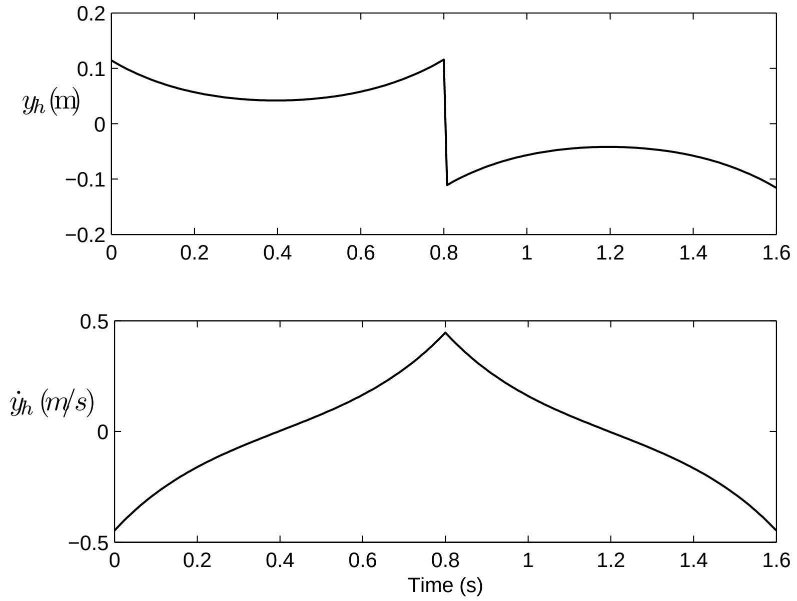


Figure 6.3: *The obtained reference hip trajectory in the frontal plane. Only two walking steps is shown*

k_p and k_v are $k_p = 7.614$, $k_v = -0.0851$. The total computation time is 664 seconds. Once the optimal value of k_p and k_v are known, the resulting optimal hip trajectory in the frontal plane can be computed based on the equations (6.1) and (6.2). Fig. 6.3 shows the obtained optimal hip trajectory in the frontal plane. The resulting ZMP trajectory in frontal plane is shown in Fig. 6.4. The thin continuous curve is the resulting ZMP trajectory obtained using the proposed method of trajectory planning. The thick continuous curve in square-wave shape is the upper bound of the stable region. The thick dashed curve in square-wave shape is the lower bound of the stable region. The two foot-centered lines are the longitudinal straight lines located at the middle of each foot. The closer the ZMP trajectory (in the frontal plane) to these foot-centered lines, the higher the stability margin . It can be seen from the figure that the resulting ZMP trajectory always stays inside the stable region and very close to the foot-centered line.

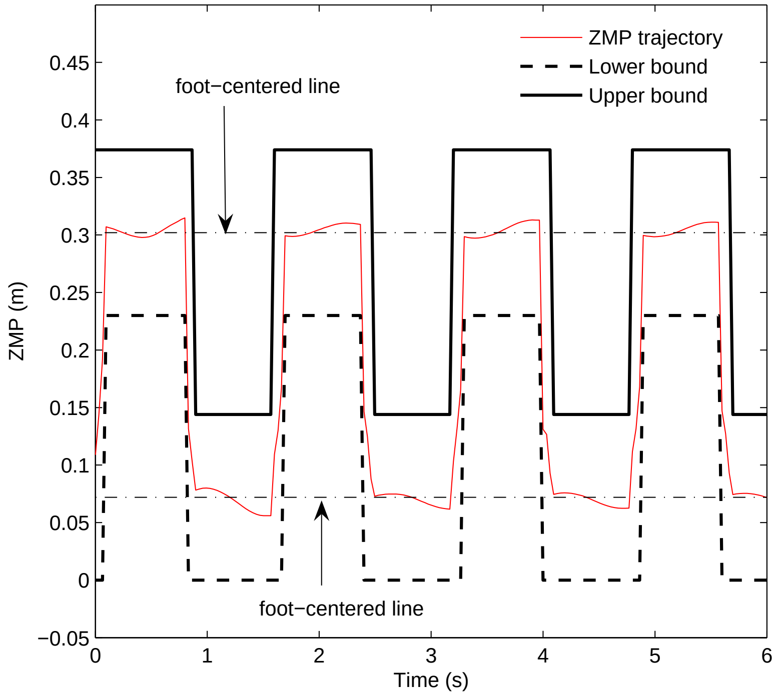


Figure 6.4: *The resulting ZMP trajectory in the frontal plane.*

This means that the obtained walking motion is stable with high stability margin.

Fig. 6.5 shows a comparison of ZMP trajectories generated using three different methods (ALIP, LIPM and GCIPM) in the frontal plane. It can be seen from the figure that the ZMP trajectory obtained using the GCIPM method stays closer to the foot-centered line compared to the ZMP trajectory obtained using the LIPM method. In addition, the ZMP trajectory obtained using the proposed ALIP method stays closer to the foot-centered line compared to that obtained using the GCIPM method. This means that the proposed method results in more stable walking trajectory compared to the LIPM and GCIPM methods.

The angle trajectories of the right hip, knee and ankle joints are shown in Fig. 6.6. The

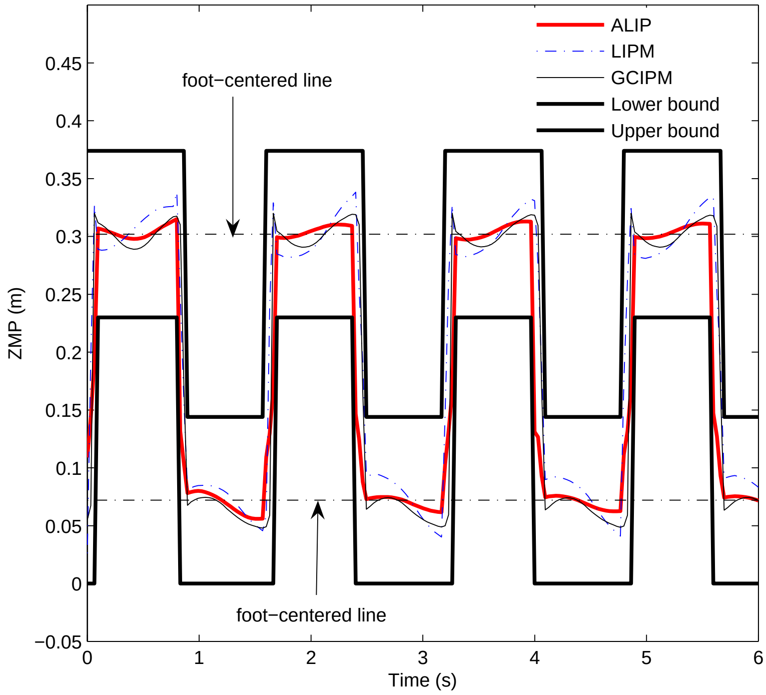


Figure 6.5: *Comparison of the resulting ZMP trajectories in the frontal plane obtained using three different methods. The continuous thick curve shows the resulting ZMP trajectory obtained using the proposed method ALIP. The continuous thin curve and dash-dotted curve are the resulting ZMP trajectories obtained using the GCIPM and LIPM methods, respectively.*

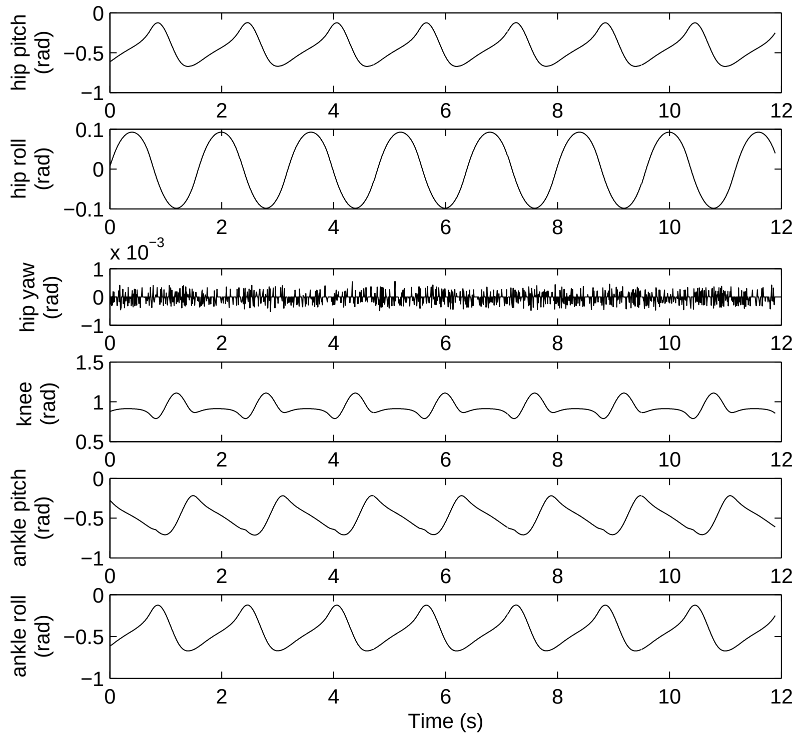


Figure 6.6: *The joint angle trajectories of the hip, knee and ankle joints of the right leg*

ground reaction force on the right foot is shown in Fig. 6.7. The upper graph shows the ground reaction forces acting on the toe and heel of the right foot while the lower graph shows the total ground reaction force (at toe and heel) acting on the foot. The resulting hip velocity is shown in Fig. 6.8. Fig. 6.9 shows the stick diagram of 3D walking motion. The images are captured at 0.08s apart.

6.2 Improve Stability Margin Using Ankle Roll Strategy

In Chapter 5 the Ankle Pitch Strategy was successfully used to improve stability margin in the sagittal plane walking. In this section, similar control strategy is applied to control

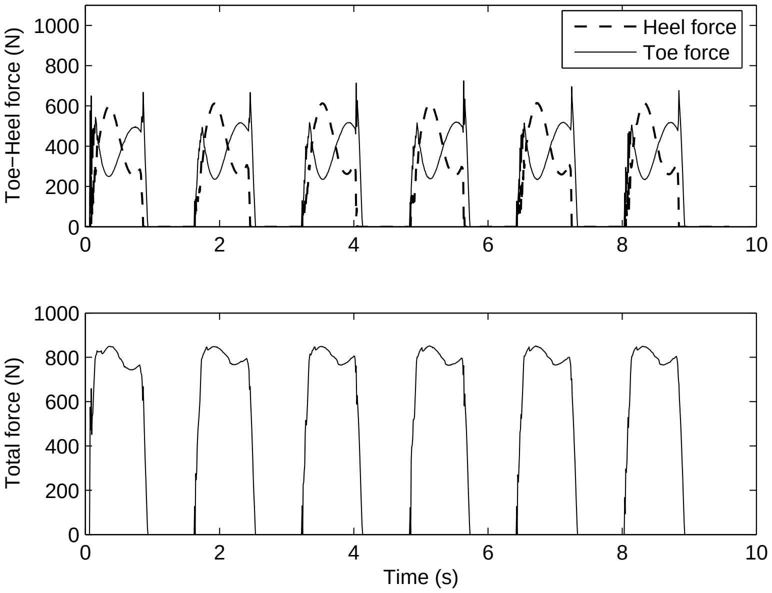


Figure 6.7: *The ground reaction force acting on the right foot*

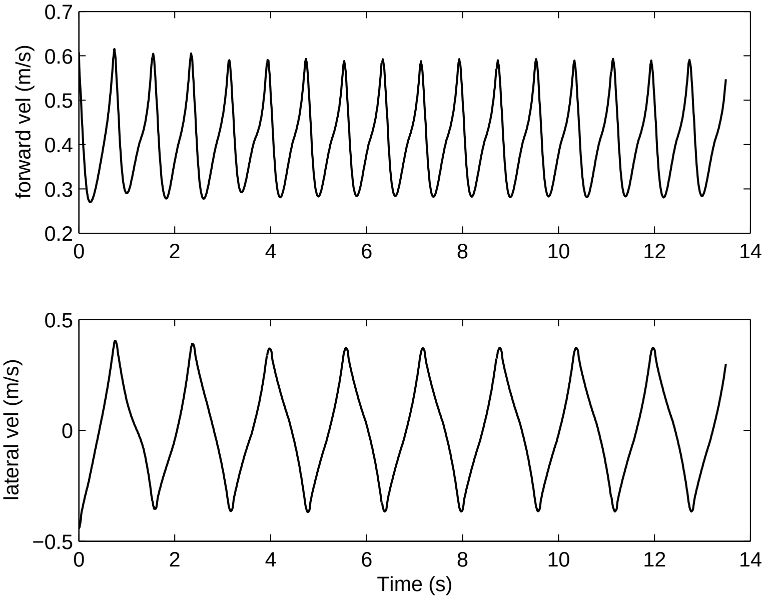


Figure 6.8: *The joint angle trajectories of the hip, knee and ankle joints of the right leg*

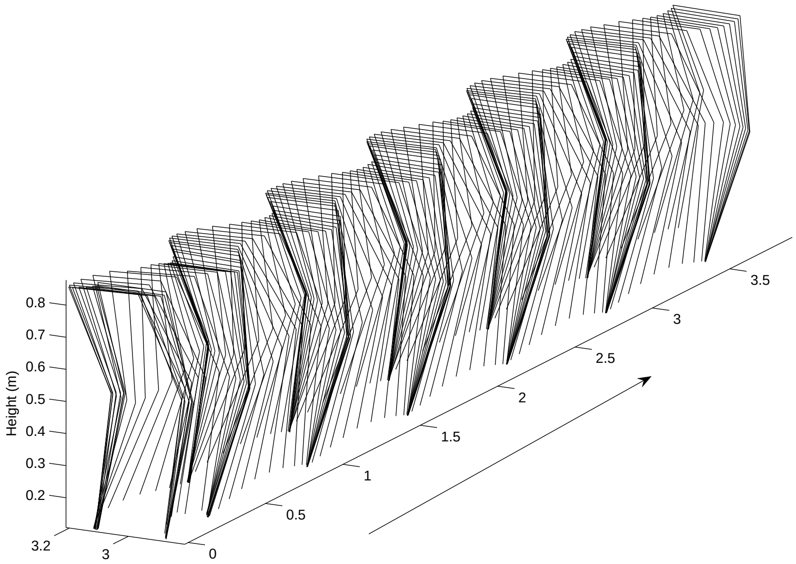


Figure 6.9: *The stick diagram of 3D walking motion. The images are captured at 0.08s apart. The direction of walking is the same as the arrow direction in the figure*

the motion in the frontal plane.

In the frontal plane, the roll angle of the ankle joint is used to control the ZMP position. The roll angle of the ankle joint is controlled based on the ground reaction force information (see Fig. 6.10). At the bottom of each foot there are four force sensors (two at the toe and two at the heel) used to measure the ground reaction force. In the sagittal plane, the reaction forces are as shown in Fig. 5.25 where F_{out} is the total reaction force at the outer toe and the outer heel and F_{in} is the total reaction force at the inner toe and the inner heel. If the reference coordinate system is placed at the outer edge of the foot as shown in Fig. 6.10, the location of the COP (or ZMP) can be determined as follows:

$$y_{ZMP} = y_{COP} = \frac{F_{in} * d_f}{F_{in} + F_{out}} \tag{6.3}$$

where y_{ZMP}, y_{COP} are the lateral location of the ZMP and COP, respectively; d_f is the width of the foot.

The control objective is to make the ZMP stays as close to the middle of the foot ($y = d_f/2$) as possiblie. This can be achieved by applying the following controller:

$$\theta_{ar} = \theta_{ar}^{ref} + \delta\theta_{ar} = \theta_{ar}^{ref} + K_{ar}(F_{out} - F_{in}) \tag{6.4}$$

where θ_{ar} is the ankle joint's roll angle to be controlled, θ_{ar}^{ref} is the reference ankle joint's roll angle computed based on the reference trajectory, $\delta\theta_{ar}$ is the compensation amount added to the ankle joint's roll angle to control the ZMP, K_{ar} is the ankle roll gain.

The controller in (5.16) is applied to the same repetitive walking simulation presented in Section 5.6.1 to check the effectiveness of the proposed controller. The ankle roll gain $K_{ar} = 0.015$ is determined by trial and error. Fig. 6.11 shows a comparison of ZMP trajectories obtained with and without applying the proposed ankle roll strategy. It can be seen from the figure that the ZMP trajectory obtained with ankle roll strategy stays closer to the foot-centered line which implies a higher stability margin achieved. This result proves the effectiveness of the proposed ankle roll controller. In this method, choosing the right gain K_{ar} is important. If the chosen gain is not good, it may lead to even worse result. However, the good news is that since there is only one parameter to be tuned, it can be easily determined using a few trial and error steps.

Simulation results show that it is possible to improve stability margin by adjusting the

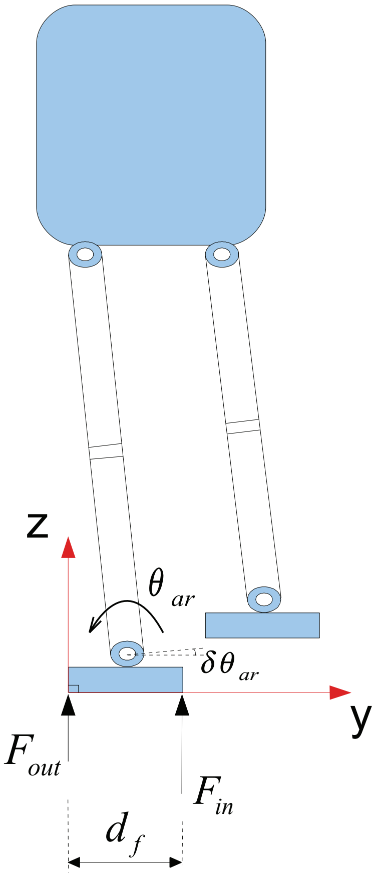


Figure 6.10: *Ankle roll compensation*

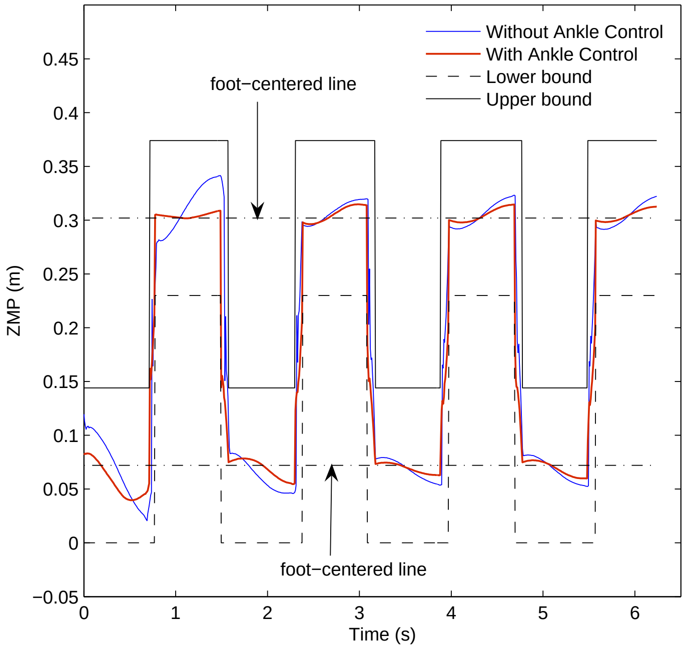


Figure 6.11: *Comparison of ZMP trajectories in two cases: with and without applying the ankle roll control strategy*

ankle joint. Although controlling the ankle joint is just a local method, it does help to improve stability of bipedal walking.

6.3 Summary

In this chapter, the 3D walking algorithms were established based on the assumption that the motions in the sagittal, frontal and transverse planes could be independently considered. The successful implementations of the algorithms indirectly validated this assumption and the proposed method.

In Section 6.1, the planning of the hip trajectory in the frontal plane was presented. The proposed model ALIP was again used to plan the hip trajectory. Genetic algorithm was also used to find the optimal parameters k_p and k_v in the frontal plane. Simulation results demonstrate that the proposed ALIP method was successfully applied for 3D walking pattern generation. In addition, the results also show that the proposed ALIP model yields better results (generate more stable walking gaits) compared to the LIPM and GCIPM models.

Section 6.2 illustrated how one could utilize the ankle joint angles to control the ZMP and hence improve stability margin. A simple controller based on the ground reaction force information was proposed to adjust the ZMP (or COP) position. The objective of the controller is to move the ZMP as close to the middle of the foot as possible. Successful implementation of the controller shows that the ankle joints can actually be used to improve walking stability.

Chapter 7

Online Walking Motion in Sagittal Plane

7.1 Introduction

In Chapter 5 and 6, the walking motions are planned offline. Offline planning means that the walking gait is planned beforehand and is then played back without any adjustment to the gait. This method is still used to generate walking motion for many humanoid robot nowadays. The offline planning method works fairly well in the condition that the walking terrain is flat, well known with little or no external disturbance affecting the biped robot. However, when the walking terrain is uneven and the external disturbance is large and uncertain, the offline walking trajectory is not suitable anymore. In this study, we aim to develop a walking algorithm that is able to adapt better to changes in walking environment. And the robot should have the ability to compensate for large external disturbances. In order to adapt well with the changes in walking environment, the robot controller must take sensor feedbacks and use it to adjust the walking gait during walking process. A walking algorithm that is able to automatically adjust the walking gaits during walking process to adapt with the changes in walking condition is called online walking algorithm.

In bipedal walking, stability of walking (or the ability to maintain balance without falling down) is probably the most important issue to be solved. There have been many bipedal research works addressing the stability issue. Bipedal researchers proposed various strategies to attain walking stability. It is widely accepted by many researchers in both robotics and biomechanics that there are generally three strategies for maintaining bipedal walking balance. For small disturbances, the strategy is to control the center of pressure (CoP) by controlling the ankle joint (therefore this strategy is also called ankle strategy). For larger disturbances where the ankle strategy is not effective anymore, the second strategy called hip strategy is used. In the second strategy, a moment is created about the center of mass (CoM) to compensate for disturbances. The compensated moment is created by controlling the hip joints therefore this strategy is called the hip strategy. For even larger disturbances, the strategy is to take an appropriate step in order to maintain balance.

In this thesis, we are interested in creating trajectories that can tolerate large disturbances. To achieve this objective it is suggested that the third strategy (taking a step) should be used. Taking an appropriate step would greatly help to capture balance. Indeed, during the process of walking, if encountered by disturbances, the robot must decide how far and how fast the swing leg should take a step in order to capture balance. If the step length is constant, the robot must take a faster step when the body's velocity is increased and a slower step when the body's velocity is decreased in order to capture balance. If the step time is constant, the robot must take a longer step when the body's velocity is increased and a shorter step when the body's velocity is decreased. Therefore, it is critical to know how fast (step time) and how far (step length) the step is so that the robot could maintain balance with highest stability margin.

7.2 Online Walking Algorithm

As mentioned in Section 7.1, in this thesis we aim to develop a walking algorithm that is able to control the desired walking speed and tolerate large external disturbances. Based on the observations of human walking and previous research works [63, 60], it is agreed that in order to maintain balance under large disturbances, the swing leg must take one or several steps to compensate for the sudden change in momentum caused by the disturbances. The key question is ”what is the best (or optimal) way to take a step?” There can be several answers to this question depending on which criteria being used. If energy consumption is the first priority then minimization of energy consumption can be used as a criterion to decide the next step of walking. If stability is the first priority then maximization of stability margin can be used as a criterion.

To decide the next walking step, there are two parameters to be determined which are step length S and step time T. There are three scenarios for determining the step length and step time. In the first scenario, the swing time is fixed and the step length is to be determined such that the walking criterion is optimized. This scenario happens when the actual foot placement or step length is not important (such as in the case of level ground walking). In the second scenario, the step length is fixed and the step time is to be determined. For this case, the actual foot placement or step length is important (such as in the case of walking on stairs or stepping stones...). In the third scenario, the two parameters step length and step time are to be determined at the same time. This scenario is more flexible and can be applied to many types of terrains. The simulation of these three scenarios are similar, therefore, in this thesis we only consider the second scenario as a demonstration of the proposed method.

In biped walking, two important state variables are the position (x) and velocity ($\dot{x}$ or v) of the center of mass (COM). Therefore, these two state variables will be used to determine the next walking step in our method. Figure 7.1 shows one step walking in the sagittal plane. The body travels from A to B according to the following equations (based on ALIP model):

$$x(t) = \begin{cases} \frac{r_2 e^{r_1 t} - r_1 e^{r_2 t}}{r_2 - r_1} x_i + \frac{e^{r_2 t} - e^{r_1 t}}{r_2 - r_1} \dot{x}_i, & \text{if } \Delta > 0 \\ [e^{rt} - rte^{rt}] x_i + te^{rt} \dot{x}_i, & \text{if } \Delta = 0 \\ [\cos(\beta t) - \frac{\alpha}{\beta} \sin(\beta t)] e^{\alpha t} x_i + \frac{\sin(\beta t)}{\beta} e^{\alpha t} \dot{x}_i, & \text{if } \Delta < 0 \end{cases} \tag{7.1}$$

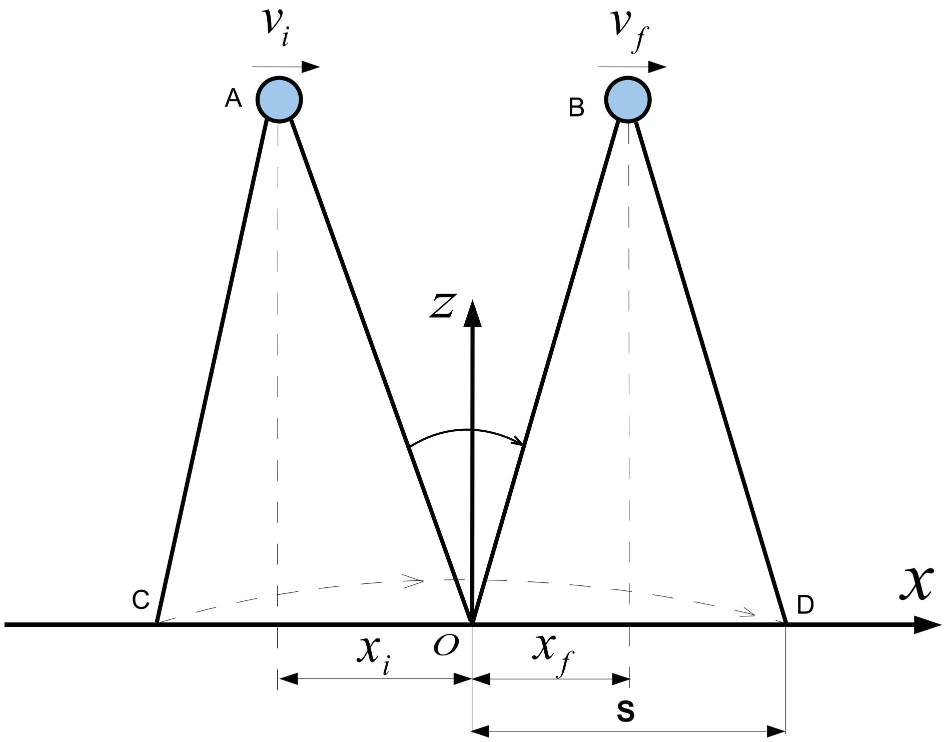


Figure 7.1: *One sample step of walking in the sagittal plane is illustrated. The body travels from A to B while the swing foot travels from C to D. x_i, v_i are the initial position and velocity of the COM, respectively. x_f, v_f are the final position and velocity of the COM, respectively. S is the step length.*

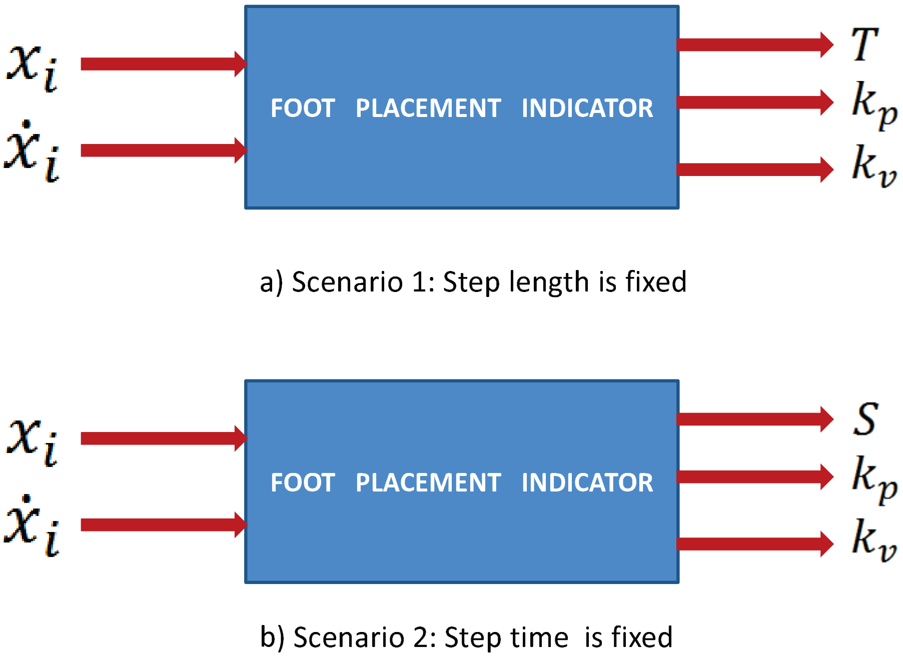
$$\dot{x}(t) = \begin{cases} \frac{r_1 r_2 (e^{r_1 t} - e^{r_2 t})}{r_2 - r_1} x_i + \frac{r_2 e^{r_2 t} - r_1 e^{r_1 t}}{r_2 - r_1} \dot{x}_i, & \text{if } \Delta > 0 \\ -tr^2 e^{rt} x_i + (1 + rt) e^{rt} \dot{x}_i, & \text{if } \Delta = 0 \\ -\frac{\alpha^2 + \beta^2}{\beta} \sin(\beta t) e^{\alpha t} x_i + [\cos(\beta t) + \frac{\alpha}{\beta} \sin(\beta t)] e^{rt} \dot{x}_i, & \text{if } \Delta < 0 \end{cases} \tag{7.2}$$

where k_p and k_v are real constants. $b = -k_v$, $c = -k_p - \frac{g}{z_o}$ and $\Delta = b^2 - 4c$; z_o is the constant height of the COM; x_i and $\dot{x}_i$ are the initial horizontal position and velocity of the COM at the start of the step, respectively; $x(t)$ and $\dot{x}(t)$ are the final horizontal position and velocity at time t.

It is desirable to have a function that is able to predict the next walking step based on the current walking states. We call such a function the **Foot Placement Indicator (FPI)**. The FPI would determine the next step based on a defined criteria. Since stability is the most critical factor in biped walking, it is chosen as the criterion guiding the Foot Placement Indicator. The function FPI must be able to determine the next step quick enough for online walking.

The inputs of the Foot Placement Indicator are the state variables x_i and $\dot{x}_i$. The outputs of the FPI are dependant on the scenarios. For instance, if the step length is fixed (scenario 1), the outputs will be the step time T and the parameters k_p, k_v. If the step time is fixed (scenario 2), the outputs will be the step length S and the parameters k_p and k_v (see Fig. 7.2).

The construction of the Foot Placement Indicator will be presented in details in the next section. For now, let's assume that the Foot Placement Indicator is available. Fig. 7.3 shows the diagram of the proposed online walking algorithm. The inputs to the Foot Placement Indicator (FPI) are the position x_i and velocity $\dot{x}_i$ of the robot's COM, which are the feedback information taken from the sensors of the robot. The outputs of the FPI are the step time T and the parameters k_p, k_v. The hip trajectory and foot trajectory of the next walking step can be determined once the input parameters (which are T, k_p, k_v, x_i and $\dot{x}_i$) are known. Once the hip and foot trajectories in Cartesian space are known, an Inverse Kinematics is required to compute the desired joint angle trajectories. And these joint angles information will be used by the position controller to control the robot. It can be seen from the diagram that the hip and foot trajectories are generated online



a) Scenario 1: Step length is fixed

b) Scenario 2: Step time is fixed

Figure 7.2: *Foot Placement Indicator*

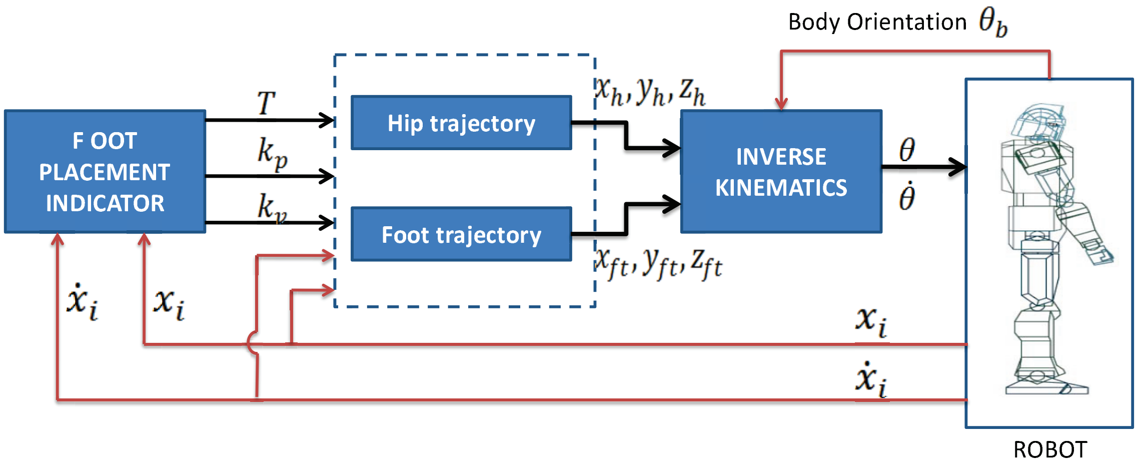


Figure 7.3: *Diagram of the Proposed Online Walking Algorithm*

based on the updated states of walking (x_i and $\dot{x}_i$). In this algorithm, the robot is able to compensate for external disturbances without knowing the magnitude, location and direction of the external disturbances. The algorithm only needs to know the position and velocity of the robot's COM to determine the next walking step.

In the next section, the construction of the Foot Placement Indicator will be presented in details.

7.3 Foot Placement Indicator (FPI)

7.3.1 Formulation of the FPI

As mentioned in the previous section, the Foot Placement Indicator (FPI) is an important part of the proposed online walking algorithm. The function of the FPI is to decide the next walking steps (how far and how fast to take the next step) during the walking process based on the current states of the biped robot. The inputs and outputs of the FPI are shown in Fig. 7.2. In order for the FPI to determine the next walking step, there must be a criterion guiding it. There are usually a few common ways to choose the guiding criterion. It can be minimization of energy consumption, or maximization of stability margin or minimization of applied joint torques or the combination of these criteria.

If energy consumption is the most important factor then the criterion can be minimization of energy consumption. If stability is the most important factor then maximization of stability margin can be used as the guiding criterion. If joint torques applied at the joints are of important concern then minimization of applied joint torque can be used as the guiding criterion.

In this study, our objective is to design a good walking trajectory that is highly stable and can compensate for large external disturbances. This suggests that stability issue would be the most important factor. Therefore, in this study, *maximization of stability margin is chosen to be the guiding criterion* of the Foot Placement Indicator.

When the step length is fixed (scenario 1), the inputs of the FPI are x_i and $\dot{x}_i$ and the outputs are T, k_p and k_v (see Fig. 7.2). Since there is no clear analytical relation between the outputs and the inputs, the optimal values of the outputs (T, k_p and k_v) will be determined using genetic algorithm (GA). The cost function of the GA algorithm is as follows:

$$\Omega = Max\{\left|x_{zmp}^{min} - d\right|, \left|x_{zmp}^{max} - d\right|\} + w\left|v_f - v_d\right| \tag{7.3}$$

where Ω is the cost function; d is the horizontal distance measured from the ankle joint to the middle point of the foot; x_{zmp}^{min}, x_{zmp}^{max} are the minimum and maximum ZMP position in one walking step, respectively (see Fig. 5.11); w is the weighting factor; v_f is the velocity of the COM at the end of walking step; v_d is the desired velocity of the COM.

Our objective is to have a trajectory that has maximum stability margin. In order to maximize stability margin, the cost function Ω must be minimized. Since in genetic algorithm, the fitness function is always being maximized, the fitness function can be formulated as follows:

$$\Phi = \frac{1}{\Omega + \delta} = \frac{1}{Max\{\left|x_{zmp}^{min} - d\right|, \left|x_{zmp}^{max} - d\right|\} + w\left|v_f - v_d\right| + \delta} \tag{7.4}$$

where δ is a constant positive number to ensure that the fitness function Φ is always definite even when the cost function Ω is zero. With the proposed fitness function, the genetic algorithm would maximize stability margin and control the velocity to a desired one (v_d) at the same time.

For each set of input (x_i, $\dot{x}_i$), GA will find a corresponding set of output (T, k_p, k_v) that maximizes the fitness function. We are interested to know whether the output is unique for a given set of input. The answer can be found by repeatedly running the GA many times (say 100 times) with different parameters used (e.g. crossover and mutation rates, population size) for a given set of input. In addition, to make sure that the algorithm does not end up with a local optimal solution, the size of the initial population should be chosen large enough. If the output is the same for all the times, it means that there is only one unique set of optimal output for a given set of input. We have done this checks for a large range of input data and found that for a given set of input, there is only one set of optimal output.

The purpose of building the Foot Placement Indicator (FPI) is that for any given set of input (x_i, $\dot{x}_i$), the FPI would be able to analytically compute the output (T, k_p and k_v). However, deriving the analytical relation between the output and the input in a direct way is impossible due to the highly complex and non-linear nature of the relation. Therefore, we propose to use interpolation technique to find the analytical relation between input and output. Each output parameter is considered as a function of the input parameters. This means that the output parameters can be described as follows: $T = f_1(x_i, \dot{x}_i)$, $k_p = f_2(x_i, \dot{x}_i)$, $k_v = f_3(x_i, \dot{x}_i)$. The functions f_1, f_2 and f_3 are constructed using interpolation technique. Since f_1, f_2 and f_3 each has two variables, we will use the Tensor Product Splines to construct these functions. Tensor Product Splines is an popular and effective way to construct two-variable functions. In the next section, the Tensor Product Splines approach will be presented.

7.3.2 Tensor Product Splines

The problem of function estimation for a two-variable function can be described as follows: given values z_r, $r = 1,...,m$, of the independent variable z, corresponding to values $(x_r, y_r) \in D$, $r = 1,...,m$, of the independent variables x and y, fit to the z_r a function $z(x,y) := z(x,y;\theta)$ of known form but containing a vector θ of n disposable parameters, to be determined such that $z(x_r, y_r) \simeq z_r$. It is noted that the functional form of $z(x,y)$ is assumed immaterial.

In this thesis, tensor product splines [12] was used to obtain function approximation. The tensor product splines is defined as follows: Consider the strictly increasing sequences $a = \lambda_0 < \lambda_1 < \cdots < \lambda_g < \lambda_{g+1} = b$ and $c = \mu_0 < \mu_1 < \cdots < \mu_h < \mu_{h+1} = d$.

The function $s(x,y)$ is called a bivariate (tensor product) spline on $R = [a,b] \times [c,d]$, of degree $k > 0$ in x and $l > 0$ in y, with knots $\lambda_i, i = 0,1,...,g+1$, in the x-direction and $\mu_j, j = 0, 1, ..., h+1$, in the y-direction. The dimension of tensor product spaces is $dim(\eta_{k,l}) = (g+k+1)(h+l+1)$. By introducing additional boundary knots $\lambda_{-k} \leq \lambda_{-k+1} \leq \cdots \leq \lambda_{-1} \leq \lambda_0 = a; b = \lambda_{g+1} \leq \lambda_{g+2} \leq \cdots \leq \lambda_{g+k} \leq \lambda_{g+k+1}$ and $\mu_{-l} \leq \mu_{-l+1} \leq \cdots \leq \mu_{-1} \leq \mu_0 = c; d = \mu_{h+1} \leq \mu_{h+2} \leq \cdots \leq \mu_{h+l} \leq \mu_{h+l+1}$ every spline $s(x,y) \in \eta_{k,l}$ then has a unique representation

$$s(x,y) = \sum_{i=-k}^{g} \sum_{j=-l}^{h} c_{i,j} N_{i,k+1}(x) M_{j,l+1}(y) \tag{7.5}$$

where the $N_{i,k+1}$ and $M_{j,l+1}$ are the normalized B-splines defined on the λ and μ knot sequences respectively. The B-splines $N_{i,k+1}$ is as follows

$$N_{i,k+1}(x) = (\lambda_{i+k+1} - \lambda_i) \sum_{j=0}^{k+1} \frac{(\lambda_{i+j} - x)_+^k}{\Pi_{\substack{l=0 \\ l \neq j}}^{k+1} (\lambda_{i+j} - \lambda_{i+l})} \tag{7.6}$$

where

$$(\lambda_{i+j} - x)_+^k = \begin{cases} (\lambda_{i+j} - x)^k, & \text{if } \lambda_{i+j} \geq x, \\ 0, & \text{if } \lambda_{i+j} < x, \end{cases} \tag{7.7}$$

The B-splines $M_{j,l+1}$ is computed in the same way as $N_{i,k+1}$.

The coefficients $c_{i,j}$ of the B-splines are determined using the Least-squares optimization method. The algorithm minimizes the following index

$$\tilde{\delta} = \sum_{q=1}^{m_1} \sum_{r=1}^{m_2} (z_{q,r} - s(x_q, y_r))^2 \tag{7.8}$$

The tensor product splines combined with the least squares optimization method will be used to compute the functions f_1, f_2, f_3 of the Foot Placement Indicator.

7.3.3 Computation of the FPI

In this section, the tensor product splines will be used to compute the output parameters of the FPI. The input parameters of the FPI are x_i and $\dot{x}_i$. The output parameters of the FPI are T (step time), k_p and k_v. For a given set of input (x_i, $\dot{x}_i$), there is a corresponding set of optimal output (T, k_p, k_v).

The relation between the output parameters and the input parameters are described as $T = f_1(x_i, \dot{x}_i)$, $k_p = f_2(x_i, \dot{x}_i)$, $k_v = f_3(x_i, \dot{x}_i)$. The functions $f_1(x_i, \dot{x}_i)$, $f_2(x_i, \dot{x}_i)$, $f_3(x_i, \dot{x}_i)$ are estimated using the function approximation method in (7.5). In order to apply the formula in (7.5), the data values of input parameters are specified at the nodes of a rectangular mesh, i.e. the data values $z_{q,r}$ is corresponding to data points (x_q, y_r), $q = 1, ..., m_1$; $r = 1, ..., m_2$.

Genetic algorithm (GA) was used to find the optimal value of the output parameters. The fitness function of the GA is as shown in (7.4). The desired velocity at the end of each step is chosen to be $\mathbf{v_d = 0.8m/s}$. For scenario 1, the step length is fixed at $\mathbf{S = 0.35m}$. The ranges of the input parameters are chosen big enough to compensate for large external disturbances. In this thesis, the ranges of x_i and $\dot{x}_i$ are chosen as follows: $-0.26 \leq x_i \leq 0.0$ and $0.3 \leq \dot{x}_i \leq 1.4$. These ranges were chosen to satisfy two conditions: (1) must be able to compensate for large disturbances and (2) maintain the gracefulness of the walking posture. The optimal value of k_p obtained using GA are presented in tables 7.1, 7.2 and 7.3. For each table, the bolded leftmost column is the value of x_i, the bolded first row is the value of $\dot{x}_i$. The rest of the data is the value of k_p corresponding to x_i and $\dot{x}_i$. For instance, in Table 7.1, when $x_i = -0.11$ and $\dot{x}_i = 0.4$, we have $k_p = 2.414$.

Since the range of the initial velocity ($0.3 \leq \dot{x}_i \leq 1.4$) is quite big, there will be cases where the difference between initial velocity and the desired velocity v_d is too big (say when $x_i = 0.3$ and $v_d = 0.8$). It is not recommended to have a trajectory where the initial velocity is too far from the desired one in one walking step. Therefore, the velocity range is divided into three parts: part 1 ($\dot{x}_i \leq 0.5$), part 2 ($0.5 \leq \dot{x}_i \leq 1.2$) and part 3 ($1.2 \leq \dot{x}_i$). Part 2 is closer to the desired velocity and part 1, 3 are farther to the desired velocity. To obtain a smooth walking motion, the following strategy is adopted:

- *if the initial velocity x_i falls into part 1 (or part 3), then it would take two steps to achieve the desired walking speed ($v_d = 0.8m/s$). The first step tries to control the velocity (v_f) to a value in part 2 (but close to part 1 (or part 3) to make sure the difference between the initial velocity and velocity at the end of each step is not too big). The second step tries to control the body velocity to the desired velocity.*

- *if the initial velocity x_i falls into part 2, it would take only one step to achieve the desired walking speed.*

To compute the parameter k_p as function of the parameters x_i and $\dot{x}_i$, the interpolation technique using tensor product splines was used. Applying Equation (7.5), the parame-

Table 7.1: Optimal value of K_p obtained when $-0.17 \leq x_i \leq 0.0$ and $0.3 \leq \dot{x}_i \leq 0.5$

		$\dot{\mathbf{x}}_\mathbf{i}$				
		0.3	**0.35**	**0.4**	**0.45**	**0.5**
$\mathbf{x}_\mathbf{i}$	**-0.17**	-4.325	-3.521	-1.971	-1.048	-0.3773
	-0.14	-3.051	-1.656	-0.4725	1.432	2.26
	-0.11	-1.264	0.895	2.414	3.86	5.051
	-0.08	2.601	4.788	6.128	8.864	9.989
	-0.05	8.988	11.94	13.26	13.6	13.86
	-0.02	14.57	22.05	22.4	20.91	19.96
	0.00	33.05	34.82	34.29	32.68	29.16

Table 7.2: Optimal value of K_p obtained when $-0.26 \leq x_i \leq 0.0$ and $0.5 \leq \dot{x}_i \leq 1.2$

		$\dot{\mathbf{x}}_\mathbf{i}$							
		0.5	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**
$\mathbf{x}_\mathbf{i}$	**-0.26**	-2.431	-0.9	1.017	3.103	4.729	5.701	6.897	6.194
	-0.23	-1.179	1.212	3.034	4.851	6.16	6.873	6.888	6.731
	-0.2	0.2527	3.513	5.271	6.634	7.801	7.943	7.44	7.518
	-0.17	2.828	5.764	7.239	8.382	8.91	8.753	8.641	8.758
	-0.155	4.475	6.56	8.165	9.282	9.641	9.619	9.597	9.766
	-0.14	6.009	7.84	9.256	9.906	10.37	10.63	10.96	10.82
	-0.125	7.231	9.099	10.36	11.13	11.56	11.99	12.14	12.17
	-0.11	8.674	9.261	11.87	12.38	12.99	13.35	13.7	14.13
	-0.095	10.03	12.21	13.58	14.06	14.66	14.8	15.59	15.83
	-0.08	12.4	14.23	15.34	16.21	16.78	17.55	18.18	19.16
	-0.065	14.76	16.66	18	18.99	19.04	19.96	20.97	22.21
	-0.05	18.18	20.01	21.2	22.2	22.74	24.33	25.45	25.34
	-0.02	28.44	30.4	31.47	32.03	32.78	34.18	34.49	35.76
	-0.00	43.65	43.98	41.77	44.14	45.82	46.26	39.08	41.04

Table 7.3: Optimal value of K_p obtained when $-0.26 \leq x_i \leq 0.0$ and $1.2 \leq \dot{x}_i \leq 1.4$

		$\dot{\mathbf{x}}_\mathbf{i}$				
		1.2	**1.25**	**1.3**	**1.35**	**1.4**
$\mathbf{x_i}$	**-0.26**	7.957	8.341	8.231	8.967	9.983
	-0.23	8.838	9.656	10.81	11.08	10.98
	-0.2	11.12	11.67	13.49	13.37	12.99
	-0.17	14.9	16	16.49	16.49	17.27
	-0.155	17.07	17.16	18.06	18.9	18.94
	-0.14	18.72	20.39	20.77	20.83	21
	-0.125	20.3	22.78	22.84	22.72	23.6
	-0.11	24.14	24.85	25.74	26.22	26.36
	-0.095	28.47	29.31	29.83	30.17	30.23
	-0.08	33.71	34.32	35.08	33.53	34.78
	-0.065	38.04	39.55	41.1	38.75	40.66
	-0.05	44.12	45.81	47.92	40.89	38.59
	-0.02	45.2	51.79	48.9	46.38	48.79
	0.00	48.59	55.62	52.41	49.42	42.02

ter can be written in the same form

$$k_p(x_i, \dot{x}_i) = \sum_{i=-k}^{g} \sum_{j=-l}^{h} c_{i,j} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{7.9}$$

where $M_{j,l+1}$ and $N_{i,k+1}$ are computed using (7.6); the coefficients $c_{i,j}$ are determined using the least-squares optimization method. Cubic splines were chosen for the B-splines (k=4, l=4).

The interpolation result for $\mathbf{k_p} = \mathbf{f_2}(\mathbf{x_i}, \dot{\mathbf{x}}_\mathbf{i})$ is follows:

- **if $\dot{\mathbf{x}}_\mathbf{i} \leq \mathbf{0.5}$ (part 1)**

$$k_p^1(x_i, \dot{x}_i) = \sum_{i=-4}^{3} \sum_{j=-4}^{1} C_{i,j}^{kp1} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{7.10}$$

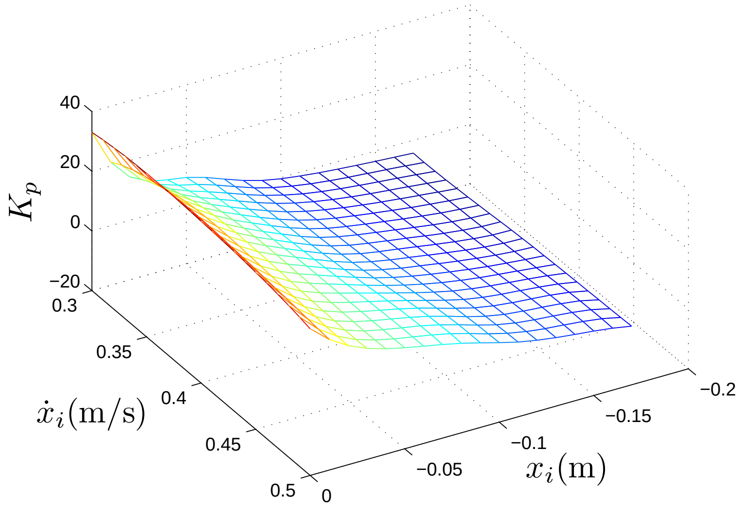


Figure 7.4: *Kp as a function of x_i and $\dot{x}_i$ (part 1)*

where

$$C_{i,j}^{kp1} = \begin{pmatrix} -4.3250 & -4.4399 & -1.2617 & -0.9207 & -0.3773 \\ -3.4950 & -1.6047 & -5.3038 & 4.5243 & 2.1657 \\ -2.4993 & -1.7503 & 4.2647 & -0.4068 & 2.2424 \\ 1.3654 & 4.8748 & 1.5194 & 12.1031 & 10.3849 \\ 12.7725 & 11.9353 & 18.1202 & 13.0058 & 14.2879 \\ 8.4961 & 23.6004 & 21.3204 & 18.2375 & 18.9124 \\ 33.0500 & 35.3817 & 34.6683 & 32.4417 & 29.1600 \end{pmatrix} \tag{7.11}$$

The plot of k_p^1 is shown in Fig. 7.4.

- **if $0.5 \leq \dot{\mathbf{x}}_\mathbf{i} \leq 1.2$ (part 2)**

$$k_p^2(x_i,\dot{x}_i) = \sum_{i=-4}^{10}\sum_{j=-4}^{4} C_{i,j}^{kp2} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{7.12}$$

where

$$C_{i,j}^{kp2} = \begin{pmatrix} -2.4310 & -1.5863 & 0.1711 & 3.1457 & 4.9498 & 5.7543 & 7.9417 & 6.1940 \\ -1.4557 & 0.2903 & 1.1622 & 4.1485 & 5.0796 & 6.5581 & 7.2294 & 6.3633 \\ -0.9356 & 1.2388 & 3.6270 & 6.0939 & 7.8632 & 8.5845 & 5.4281 & 7.1631 \\ 1.6713 & 5.3812 & 6.5587 & 7.3747 & 8.6375 & 7.6700 & 8.5474 & 8.0188 \\ 4.0489 & 5.8284 & 7.0435 & 9.6696 & 9.5258 & 9.4380 & 8.7372 & 9.5081 \\ 6.0401 & 6.5085 & 9.5773 & 9.4557 & 10.3296 & 10.3251 & 11.5247 & 10.7244 \\ 7.0963 & 10.1504 & 9.5242 & 11.4125 & 11.3929 & 12.3529 & 11.8433 & 11.9291 \\ 8.8036 & 4.8199 & 13.0808 & 11.9136 & 13.1256 & 13.5789 & 13.6657 & 14.3774 \\ 9.7274 & 12.7970 & 13.6799 & 13.8698 & 14.8832 & 13.6947 & 16.5864 & 15.3311 \\ 12.7255 & 13.7913 & 14.8094 & 16.3106 & 16.8583 & 18.8296 & 18.0967 & 19.6332 \\ 14.9822 & 16.8023 & 18.1233 & 20.1442 & 18.7768 & 20.1772 & 21.6188 & 22.8228 \\ 22.2938 & 23.5696 & 23.7421 & 25.1803 & 27.0873 & 32.2310 & 31.4982 & 28.2469 \\ 28.6891 & 29.9498 & 34.6247 & 33.2311 & 33.0643 & 34.1135 & 36.8177 & 39.5136 \\ 43.6500 & 47.2123 & 38.9252 & 44.7093 & 45.4627 & 49.5008 & 31.4771 & 41.0400 \end{pmatrix} \tag{7.13}$$

The plot of k_p^2 is shown in Fig. 7.5.

- **if $\dot{\mathbf{x}}_\mathbf{i} \geq \mathbf{1.2}$ (part 3)**

$$k_p^3(x_i, \dot{x}_i) = \sum_{i=-4}^{10} \sum_{j=-4}^{1} C_{i,j}^{kp3} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{7.14}$$

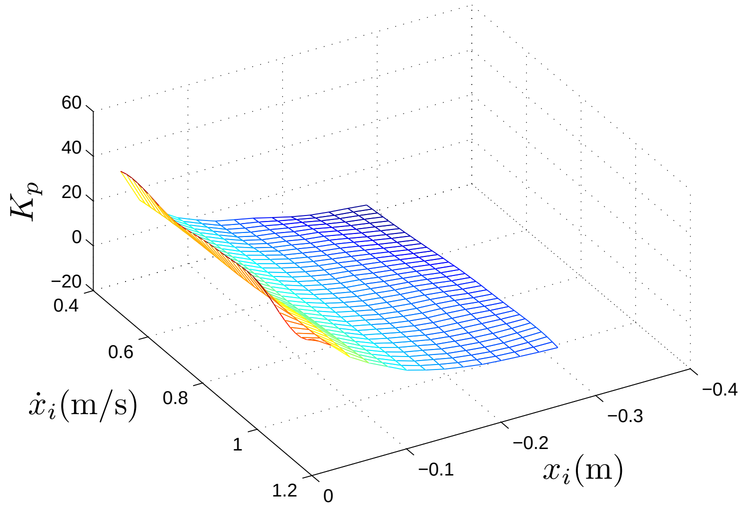


Figure 7.5: *Kp as a function of x_i and $\dot{x}_i$ (part 2)*

where

$$C^{kp3}_{i,j} = \begin{pmatrix} 7.9570 & 8.7813 & 7.3493 & 9.4440 & 9.9830 \\ 8.0675 & 12.4290 & 9.6785 & 9.6201 & 11.3167 \\ 9.4007 & 2.7864 & 14.6557 & 12.8726 & 9.9143 \\ 13.2441 & 17.2523 & 17.2937 & 12.8513 & 16.3351 \\ 16.6716 & 14.7269 & 16.6929 & 19.4530 & 18.4070 \\ 18.8278 & 20.5103 & 21.2811 & 20.6789 & 20.6744 \\ 19.6092 & 24.0120 & 22.1707 & 21.9380 & 23.6400 \\ 24.2654 & 23.1174 & 26.4740 & 26.1436 & 26.0331 \\ 28.1608 & 29.8736 & 28.6581 & 31.4567 & 30.3961 \\ 34.5469 & 32.9975 & 39.2186 & 30.0745 & 34.2328 \\ 38.2142 & 38.8518 & 45.6566 & 38.8221 & 44.7736 \\ 53.1822 & 44.7952 & 77.9440 & 26.2867 & 28.2728 \\ 40.2251 & 61.0931 & 30.2047 & 54.2850 & 62.6457 \\ 48.5900 & 39.8528 & 49.7485 & 50.2906 & 42.0200 \end{pmatrix} \tag{7.15}$$

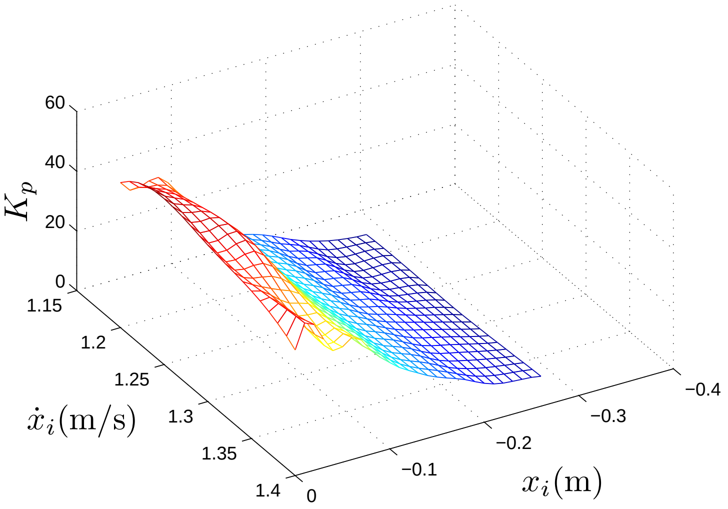


Figure 7.6: *Kp as a function of x_i and $\dot{x}_i$ (part 3)*

The plot of k_p^3 is shown in Fig. 7.6.

The results of the functions $\mathbf{T(x_i, \dot{x}_i)}$ and $\mathbf{k_v(x_i, \dot{x}_i)}$ are presented in the Appendix.

7.4 Simulation Results

7.4.1 Online Level Walking With No Disturbance

In this section, simulation results of the online level walking in the sagittal plane is presented. In this experiment, it is assumed that there is no disturbance during the process of walking. The specifications of the simulated biped were taken from the biped HUBIRO(see Fig. 1.1). Yobotics (http://yobotics.com/), a dynamic simulator, was used to simulate the walking algorithm.

The simulation is done with the following input parameters: the desired step length is $\mathbf{S = 0.35m}$ (scenario 1), the desired velocity at the end of each step is $\mathbf{v_d = 0.8m/s}$. Since the step length is fixed, the inputs to the Foot Placement Indicator are x_i and $\dot{x}_i$ and the outputs to be determined are the step time T_i and the parameters k_p and k_v. Applying the proposed algorithm shown in Fig. 7.3, the online walking motion was successfully achieved. Firstly, the Foot Placement Indicator (FPI) function is built in advance by using the Tensor Product Spline interpolation technique. The construction of the FPI function is done off-line. The FPI function is in analytical form, therefore, it can give the output almost instantaneously which is suitable for real-time simulation.

Fig. 7.7 shows the simulation data of the obtained online walking motion where T_i is the time of one walking step computed by the FPI, $\dot{x}_{hip}$ is the resulting velocity profile of the hip, x_{hip}^{ref} is the computed reference hip trajectory which is updated every walking step, x_i is the measured position of the COM at the beginning of each step, $\dot{x}_i$ is the measured velocity of the COM at the beginning of each step. It can be seen from the figure that the velocity at the end of each step (v_f) is close to the desired velocity $(v_f^d = 0.8m/s)$.

Unlike the off-line walking motion where the reference trajectory is fixed, in this online walking simulation the reference trajectory of each step is computed based on the feedback information (x_i and $\dot{x}_i$) therefore the reference trajectory is more adaptive to environment changes. The step time T_i of each step i^{th} is also changing from step to step depending on the states of walking. Fig. 7.8 shows the resulting ZMP trajectory of the obtained walking motion. It can be seen from the figure that the ZMP trajectory stays very close to the middle of the stable region which means that the stability margin is very close to the maximum value. Therefore, it can be concluded that the obtained

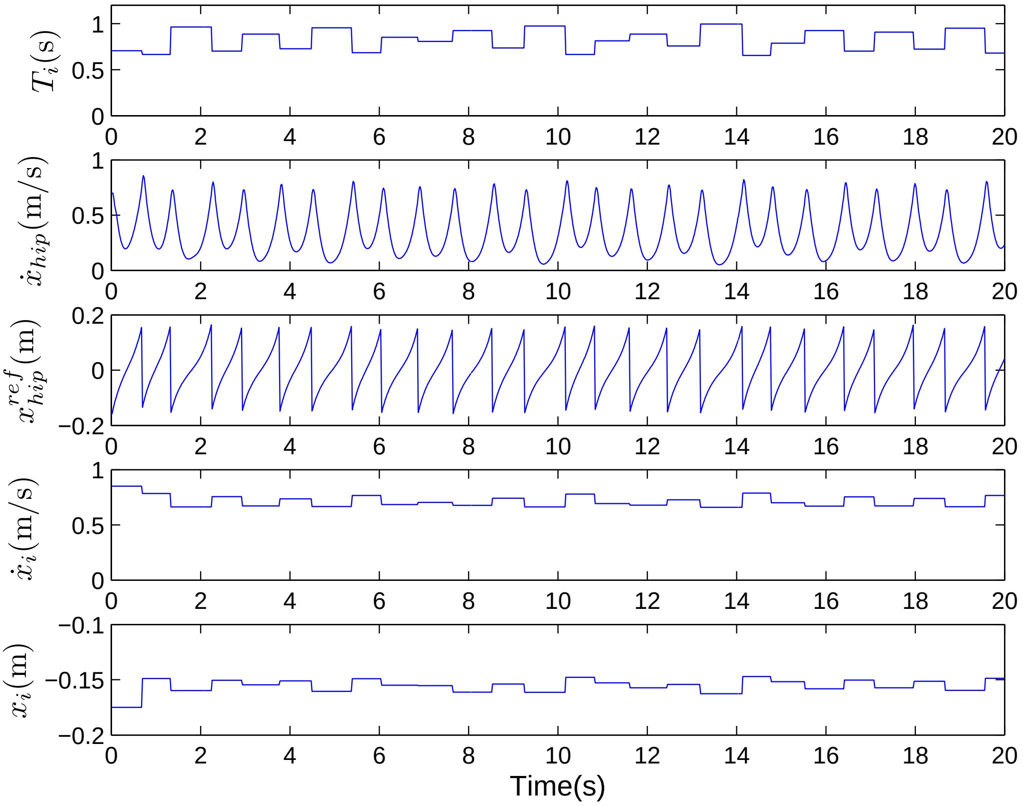


Figure 7.7: *The obtained simulation data for online walking motion without external disturbance*

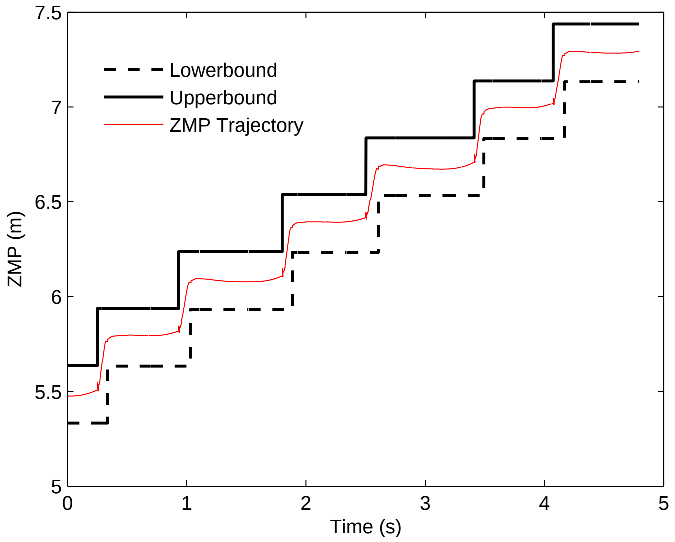


Figure 7.8: *The resulting ZMP trajectory of the online walking simulation. The thick solid curve is the upper bound of the stable region, the dashed curve is the lower bound of the stable region and the thin solid curve is the ZMP trajectory.*

walking motion has a high degree of stability. Fig. 7.9 shows the joint-angle trajectories of the online walking robot. The stick diagram of the walking motion is shown in Fig. 7.10.

7.4.2 Online Level Walking Under Disturbance

In the previous section, the effectiveness of the proposed online walking algorithm was presented through the simulation results of walking motion on flat terrain without external disturbance. In this section, to examine the robustness of the proposed walking algorithm, the walking robot is put under large external disturbance.

The input parameters are kept the same as in the previous section i.e. the desired step length is $\mathbf{S = 0.35m}$, the desired velocity at the end of each step is $\mathbf{v_d = 0.8m/s}$. Let's

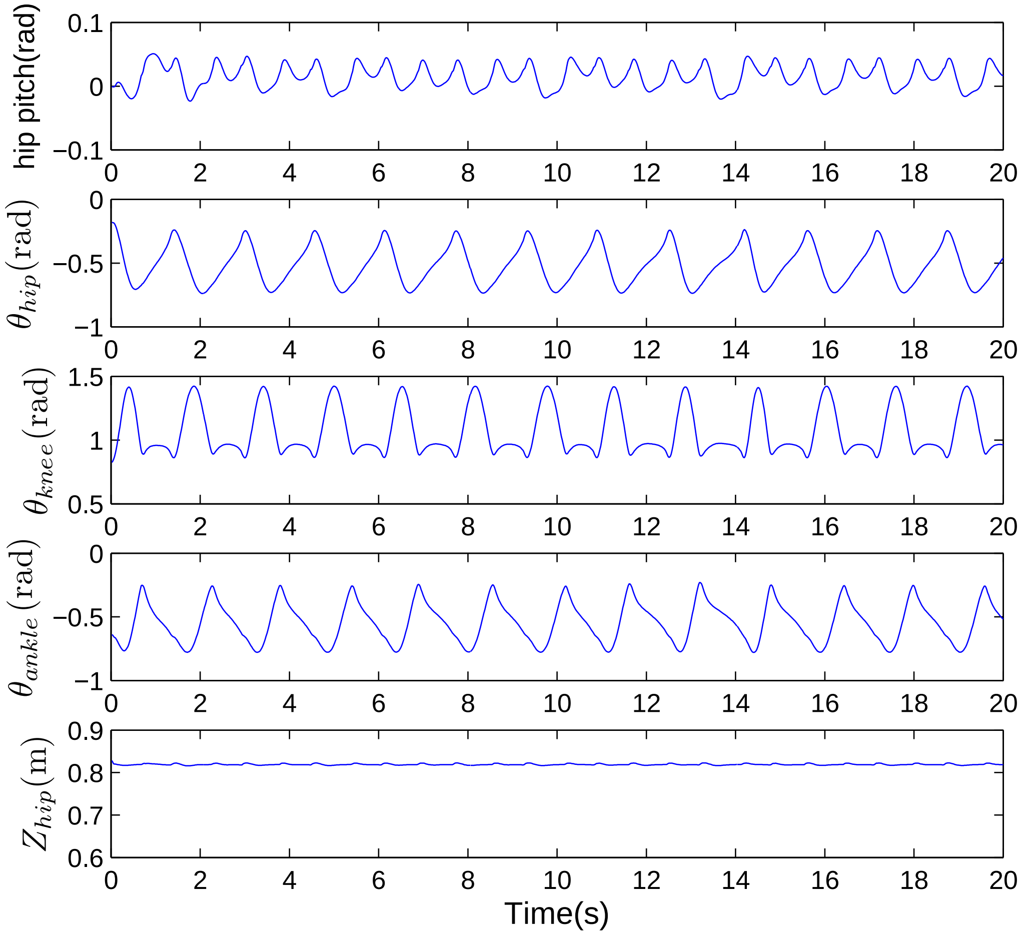


Figure 7.9: *The joint-angle trajectories of the online walking robot*

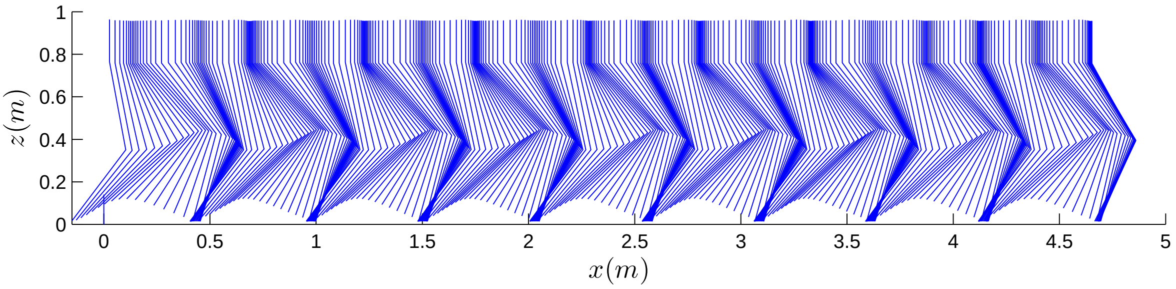


Figure 7.10: *The stick diagram of the obtained online walking motion. The images are captured at 0.04 s apart. Only the right leg is shown.*

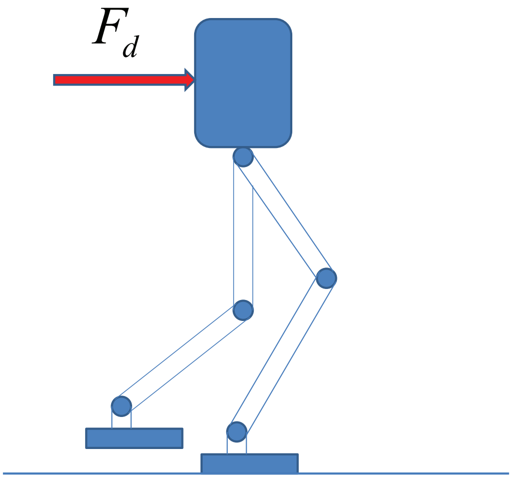


Figure 7.11: *Disturbance force F_d acting on the robot's trunk*

call the external disturbance force F_d acting on the robot's trunk in the duration of ΔT_d seconds (Fig. 7.11). For a given value of ΔT_d, there is a maximum value of F_d that the robot can tolerate without falling down. Table 7.4 shows the maximum value of the disturbance force allowed for various time period ΔT_d. From the table, it can be seen that the longer the disturbance time, the smaller the disturbance force the walking algorithm can compensate. This suggests that disturbance effect on a walking robot not only attributed to the magnitude of disturbing force but also the duration of disturbance. In other words, disturbance effect is caused by a combination of disturbing force and the duration of disturbance (Disturbance effect = $F_d * \Delta T_d$ = Change of linear momentum). Therefore, to determine whether a disturbance is large or small we use *the change in linear momentum* as a measure. A disturbance is considered ”large” if the change in momentum caused by the disturbance is equal or greater than 50% of the walking robot's momentum. In this section, detailed results of the case when $\Delta T_d = 0.5s$ will be presented.

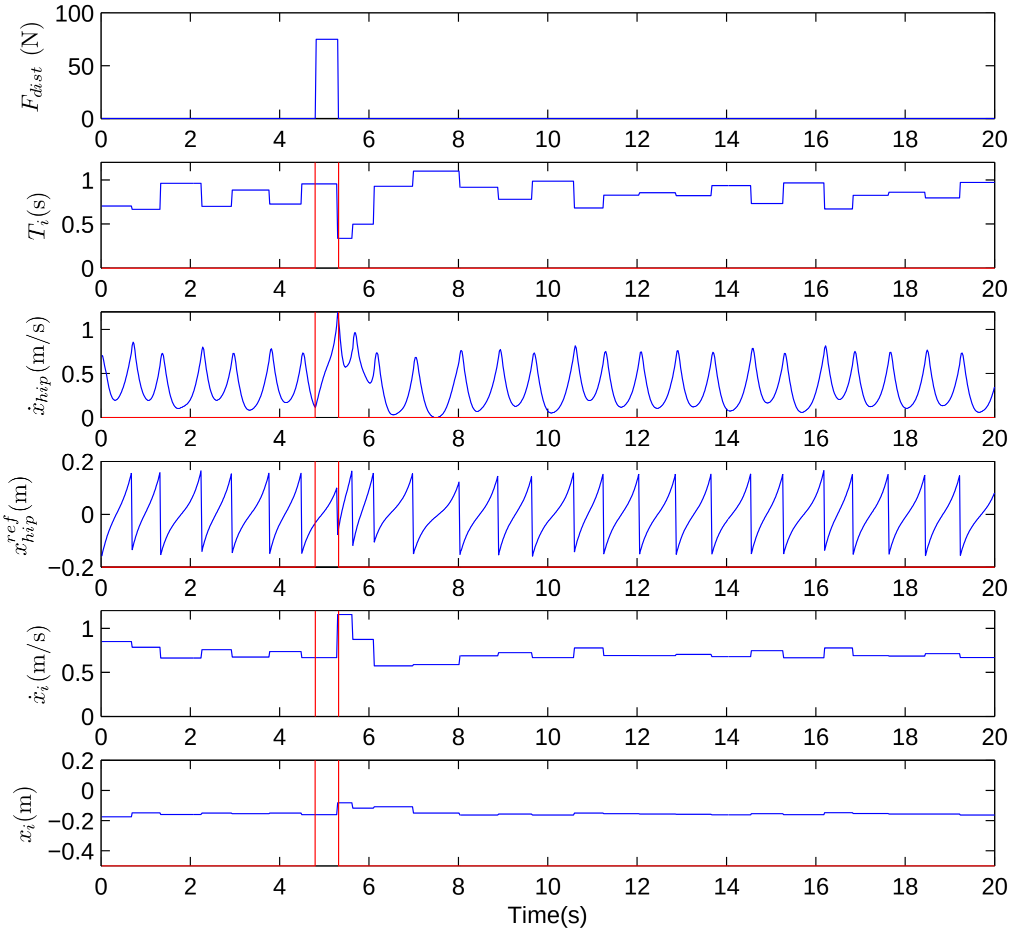


Figure 7.12: *The obtained simulation data for online walking motion under external disturbance $F_d = 75N$ during a period of $\Delta T_d = 0.5s$.* The duration of disturbance is indicated using two vertical lines as shown in the figure

Table 7.4: Maximum disturbance force F_d^{max} allowed for different period of time ΔT_d

ΔT_d	0.1	0.2	0.3	0.4	0.5
F_d^{max}	310	165	115	92	75

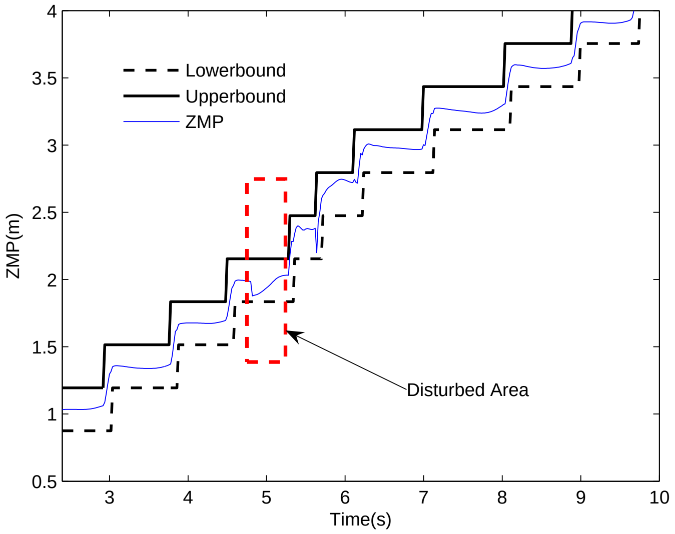


Figure 7.13: *The resulting ZMP trajectory of the biped walking robot under disturbance*

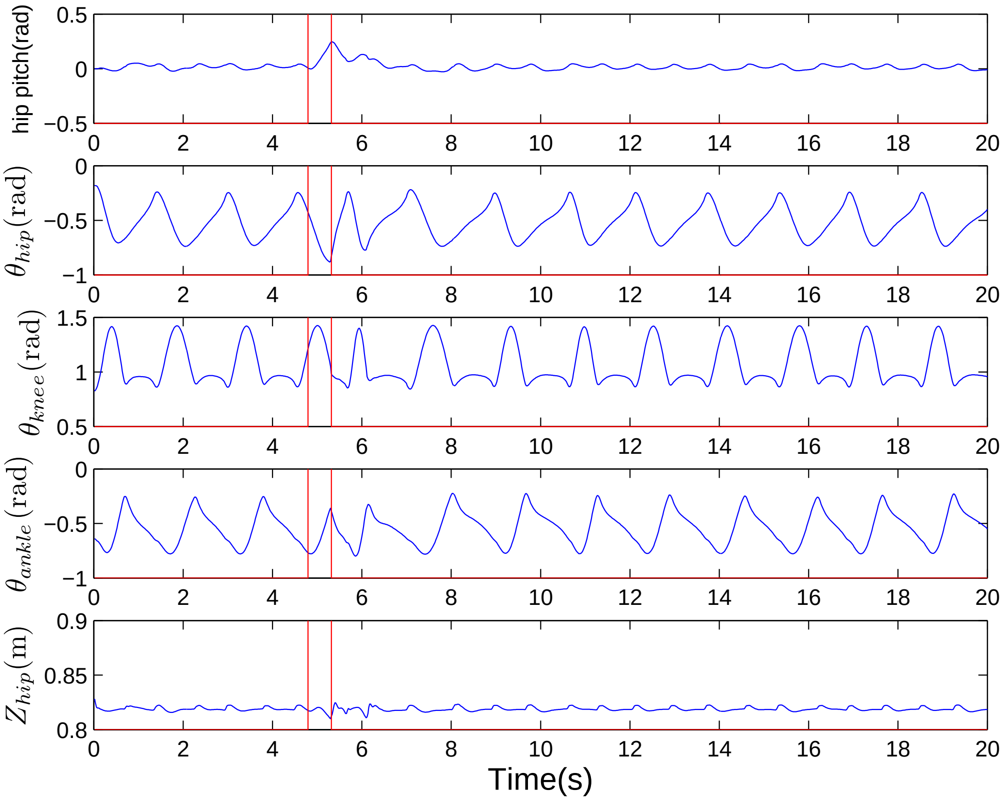


Figure 7.14: *The resulting joint angle trajectories of the biped robot under disturbance*

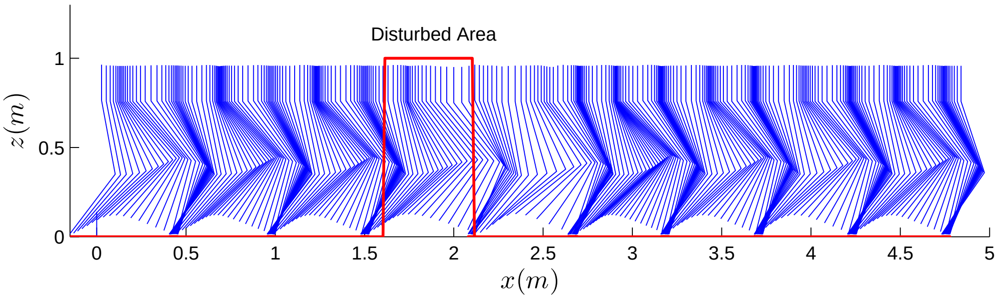


Figure 7.15: *The stick diagram of the online walking motion under large disturbance. Images are captured at 0.04s apart. Online the right leg is shown.*

As shown in table 7.4, when $\Delta T_d = 0.5s$ the maximum disturbance force that the robot can compensate is $F_d^{max} = 75N$. This means that the proposed walking algorithm can compensate for the change in momentum (caused by the disturbance) of $\Delta P_d = F_d * \Delta T_d = 75 * 0.5 = 37.5Nm/s$. Whereas the momentum of the walking robot is $P = m * \bar{v} = 86.59 * 0.4 = 34.63Nm/s$ ($\bar{v} = 0.4m/s$ is the average velocity). We have $\Delta P_d/P = 37.5/34.63 = 1.08 = 108\%$. This means that the change in momentum caused by the external disturbance is 108% times greater than the robot's momentum. Therefore, we can conclude that the proposed walking algorithm can compensate for fairly large external disturbance. Fig. 7.12 shows the obtained simulation data for online walking motion under external disturbance $F_d = 75N$ during a period of $\Delta T_d = 0.5s$. It can be seen from the figure that right after the disturbance the velocity increases almost double and the stepping time T_i reduces sharply (almost one half). This result makes sense because after disturbance the velocity increases a lot and when the body's velocity increases the robot must take faster step (shorter step time) in order to catch up with body's velocity. It can also be referred from the figure that after about two walking steps the robot is able to return to desired velocity profile.

Figure 7.13 shows the resulting ZMP trajectory of the biped walking under disturbance. It can be seen that the ZMP always stay inside the stable region and the stability margin reduces during and just after disturbance. After disturbance the robot is able to maintain a high stability margin walking motion as can be seen from the figure. Fig. 7.14 shows the joint angle trajectories of the bipedal walking under disturbance. The stick diagram of the online walking motion under disturbance ($F_d = 75N$, $\Delta T_d = 0.5s$) is shown in Fig. 7.15. From this figure we can see clearly how the walking gait changes when disturbance occurs.

7.5 Summary

In this chapter, the proposed Augmented Linear Inverted Pendulum (ALIP) model was used to generate online walking motion. Our objective is to generate a robust walking algorithm that is able to regulate walking speed and tolerate large disturbances. To serve this objective, a function called Foot Placement Indicator (FPI) was proposed. The role of the FPI is to decide the next walking step based on current walking states so that the walking robot could maintain balance and walking speed even under large disturbance. The inputs of the FPI are the position (x_i)and velocity ($\dot{x}_i$) of the COM at the beginning of each step. For the outputs of the FPI, there are two cases. If the step length is fixed, the outputs are the step time T, and the parameters k_p, k_v. If the step time is fixed, the outputs are the step length S and the parameters k_p, k_v. The FPI is constructed using the Tensor Product Splines combined with the least square optimization method. Each of the output parameters are considered as a function of the input parameters (x_i and $\dot{x}_i$).

The proposed online walking algorithm was successfully applied to the simulation of a life-sized biped robot named HUBIRO in sagittal plane. The first simulation was done for normal walking condition on flat terrain. Simulation results show that a smooth and highly stable walking motion was obtained. For each walking step, the FPI quickly compute the optimal value of the parameters k_p, k_v and the step time T based on the walking states x_i, $\dot{x}_i$. In the second simulation, the walking robot was put under large external disturbance to test the robustness of the algorithm. An external disturbance force F_d was applied to the COM of the robot from behind in a period of ΔT_d seconds. Simulation results show that the proposed walking algorithm is able to compensate for a maximum disturbance force $F_d = 75N$ applied in a period of $\Delta T_d = 0.5s$. This disturbance is considered large because it causes a 108% change in linear momentum of the walking robot. This disturbance caused a double increase in velocity at the end of disturbed period. Simulation result shows that the biped robot adapts well with this disturbance by computing the appropriate stepping time to capture balance and quickly bring the robot to normal walking condition (after one walking step). When the disturbed time is reduced to $\Delta T_d = 0.1s$, the proposed walking algorithm is able to compensate for a maximum disturbance force of $F_d = 310N$.

Chapter 8

Conclusions and Future Works

8.1 Conclusions

The ultimate purpose of this thesis is to introduce a simple yet effective dynamic model for biped walking. One popular simple biped model used widely by many researchers is the Linear Inverted Pendulum Mode (LIPM). Unfortunately, in this model the dynamics of the legs (arms) was not considered. This thesis tried to analyze the effects of the dynamics of the legs on the whole robot dynamics. Simulation results show that the effect of the legs is significant especially when the leg's mass is big or when the swinging speed is high.

This thesis proposed a new model for bipedal walking called ”The Augmented Linear Inverted Pendulum (ALIP)”. In this model, an augmented function F is added to the dynamic equation of the Linear Inverted Pendulum. The role of the augmented function is to improve the inverted pendulum dynamics by indirectly incorporating the dynamics of the arms, legs, heads, etc into the dynamics equation. The inverted pendulum dynamics can be easily adjusted or modified by changing the key parameters of the augmented function. Genetic algorithm is used to find the optimal value of the key parameters of the augmented function. Our objective is to design a walking pattern that has the highest stability margin possible. Stability margin is computed based on the Zero-moment-point (ZMP) information. The proposed ALIP model was used to generate off-line walking pattern for biped robot in 2D and 3D walking. Simulation results show that the proposed ALIP model is able to generate highly stable walking patterns. The walking patterns generated using the proposed approach is more stable than that generated using the LIPM model and GCIPM (an improved version of the LIPM).

To further improve the stability margin, an ankle control strategy was proposed. In this strategy, the ankle joint is controlled such that the ZMP stays as close to the middle point of the supporting foot as possible. This is obtained by adjusting the ankle pitch and roll angles based on the ground reaction force information so that the difference between the ground reaction force at the heel and toe is minimized. Simulation results show that the proposed method is effective in increasing the stability margin of the bipedal walking robot.

The proposed ALIP model was also successfully applied to generate online walking motion in sagittal plane. The online walking algorithm comprises of a proposed function called the Foot Placement Indicator (FPI). The Foot Placement Indicator (FPI) is an important part of the online walking algorithm. The role of the FPI is to decide the next walking steps (how far and how fast to take the next step) during the walking process based on the current states of the biped robot (position x_i and velocity $\dot{x}_i$ of the COM). The inputs of the FPI are the walking states x_i, $\dot{x}_i$ and the outputs are the key parameters k_p, k_v and the step time T (if the step length is fixed) and the step length (if the step time is fixed). Each of the output parameters is constructed as a function of the inputs x_i, $\dot{x}_i$ using the tensor product spline interpolation method. Simulation results show that the obtained walking motion is highly stable and adaptive to the changes in walking conditions. The simulated biped robot is able to compensate for fairly large disturbance force (disturbance force $F_d = 75N$ applied in a period of $\Delta T_d = 0.5s$ - this disturbance caused a 108% change in linear momentum of the walking robot which is considered big). The proposed online walking algorithm is able to compensate for large external disturbance because the foot placement is computed regularly based on the two important walking states x_i and $\dot{x}_i$. Indeed, when the foot placement of the next step is decided based on the current position and velocity of the COM, the robot is able to place the swing foot at the right time and right place to ”catch up” with the increase (or decrease) in walking speed and hence compensating for the change in linear momentum. This simulation result further confirms the point that taking an appropriate step is an effective way to maintain balance when encountered large disturbances.

The proposed method ALIP was proved to generate more stable walking pattern than that generated using the LIPM and GCIPM methods. However, no method is perfect and the ALIP method is not exceptional. The disadvantage of the proposed method is that when the walking parameters (such as step length, step time) or the physical properties of the robot (such as mass distribution, inertia moments) are changing the optimization process needs to be run again. This is reasonable because it is impossible to have one unique optimal solution problems with different inputs.

8.2 Future Works

Based on the results of this study, there are still several things to be done to make the complete picture of the proposed method. In this section, we propose some of the possible works to be done in the future.

The ultimate goal of bipedal walking researchers is to build robots that can walk on various types of terrains like human walking. This means that the robots must be able to walk on flat terrains, uneven terrains, staircases, slopes, etc. In this thesis, all of the simulations were done on flat terrain so far. It is a natural extension to add the uneven terrain walking simulation to further test the effectiveness and robustness of the proposed method. For uneven terrain walking, the ankle joint must be able to adapt quickly to the sudden change of the walking surface. Using a compliant ankle joint may help to increase stability for walking robot.

Besides the rough terrain walking, the biped should also have the capability to change its walking direction while walking. All of the simulations in this thesis so far only consider the case of walking on a straight line. In order for the robot to operate and implement complex tasks it is a must that the robot is able to control its walking direction.

So far, 3D walking simulation was achieved for off-line walking. For online walking motion, only 2D simulation was reported. It is necessary to extend this online algorithm to 3D walking. The extension should be straightforward and similar to 2D walking. The same algorithm used for sagittal plane can be applied for frontal plane. For 3D online walking, however, will be more challenging because the motion in the sagittal plane may affect the motion in the frontal plane if the two motion are not well coordinated.

The Foot Placement Indicator (FPI) is an important part of the online walking algorithm. The role of the FPI is to determine the next walking step (how far and how fast to take the next step so that the walking motion has the highest stability). This thesis only considers the cases when either the step length of the step time are fixed. It should be interesting to consider the case when both of the step length and step time are not fixed. This means that the FPI would determine both step length and step time for the next walking step concurrently.

Bibliography

[1] M. Abdallah and A. Goswami. A biomechanically motivated two-phase strategy for biped upright balance control. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005 (ICRA 2005)*, pages 1996 – 2001, 2005.

[2] Arakawa and Fukuda. Natural motion trajectory generation of biped locomotion robotusing genetic algorithm through energy optimization. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1495 – 1500, 1996.

[3] H. Benbrahim and J. A. Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22:283–302, 1997.

[4] G. Capi, S. Kaneko, K. Mitobe, L. Barolli, and Y. Nasu. Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. *Robotics And Autonomous Systems*, 38:119–128, 2002.

[5] C. M. Chew. *Dynamic bipedal walking assisted by learning*. PhD thesis, Massachusetts Institute of Technology, September 2000.

[6] C. M. Chew and G. A. Pratt. Dynamic bipedal walking assisted by learning. *Robotica*, pages 447–491, 2002.

[7] S. H. Choi, Y. H. Choi, and J. G. Kim. Optimal walking trajectory generation for a biped robot using genetic algorithm. In *Proceedings. 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. (IROS '99)*, volume 3, pages 1456 – 146, 1999.

[8] H. Dau, C. M. Chew, and A. N. Poo. Achieving energy-efficient bipedal walking trajectory through ga-based optimization of key parameters. *International Journal of Humanoid Robotics (IJHR)*, 6:609–629, 2009.

[9] V. H. Dau, C. M. Chew, and A. N. Poo. Optimal trajectory generation for bipedal robots. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, page 603, 2007.

[10] V. H. Dau, C. M. Chew, and A. N. Poo. Optimized joint-torques trajectory planning for bipedal walking robots. In *2008 IEEE Conference on Robotics, Automation and Mechatronics*, page 1142, 2008.

[11] V. H. Dau, C. M. Chew, and A. N. Poo. Proposal of augmented linear inverted pendulum model for bipedal gait planning. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan, 2010.

[12] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, 1995.

[13] Y. J. Ellenberg and R. Oh. Realization of miniature humanoid for obstacle avoidance with real-time zmp preview control used for full-sized humanoid. In *10th*

IEEE-RAS International Conference on Humanoid Robots (Humanoids), page 46, 2010.

[14] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng. An empirical exploration of a neural oscillator for biped locomotion control. In *IEEE International Conference on Robotics and Automation*, pages 3036–3042. New Orieans, April 2004.

[15] J. Ferreira, M. Crisostomo, and A. Coimbra. Zmp trajectory reference for the sagittal plane control of a biped robot based on a human cop and gait. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, page 1588, 2009.

[16] Y. Fujimoto and A. Kawamura. Proposal of biped walking control based on robust hybrid position force control. In *Proceedings of the IEEE International Conference Robotics and Automation*, pages 2724–2730, 1996.

[17] D. Goldberg. *Genetic algorithm in search optimization and machine learning*. Addison-Wesley, 1989.

[18] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Proceeding of the IEEE Conference on Robotics and Automation*, pages 3785 – 3790, 2004.

[19] S. Grillner. *Control of locomotion in bipeds, tetrapods, and fish*. The Nervous System. Handbook of Physiology, 1981.

[20] S. Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228:143–149, 1985.

[21] Hayes and Demiris. robot controller using learning by imitation. In *International Symposium on Intelligent Robotic Systems*, pages 198–204. Grenoble.

[22] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proc. 1998 ICRA*, pages 1321–1326, 1998.

[23] F. Horak and L. Nashner. Central programming of postural movements: adaptation to altered support surface configurations. *Journal of Neurophysiology*, 55:1369–1381, 1986.

[24] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie. Planning walking patterns for a biped robot. In *IEEE Transactions on Robotics and Automation*, volume 17, 2001.

[25] W. W. Huang, C. M. Chew, and G. S. Hong. Neural oscillator structure for rhythmic control of human locomotion. In *International Conference on Climbing and Walking Robots*, 2007.

[26] W. W. Huang, C. M. Chew, and G. S. Hong. Coordination between oscillators: An important feature for robust bipedal walking. In *IEEE Int Conf on Robotics and Automation (ICRA) 2008*, pages 3206 – 3212, 2008.

[27] W. W. Huang, C. M. Chew, Y. Zheng, and G. S. Hong. Bio-inspired locomotion control with coordination between neural oscillators. *International Journal of Humanoid Robotics*, 6(4):585–608, 2009.

[28] T. Inamura, I. Toshima, and N. Yoshihiko. *Acquiring motion elements for bi-*

directional computation of motion recognition and generation, volume 372381 of *Experimental Robotics VIII*. Springer, 2003.

[29] K. S. Jeon, O. Kwon, and J. H. Park. Optimal trajectory generation for a biped robot walking a staircase based on genetic algorithms. In *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004)*, volume 3, pages 2837 – 2842, 2004.

[30] S. Kajita, F. Kanechiro, Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings of the 2001 IEEE/RSI International Conference on Intelligent Robots and Systems*, volume 1, pages 239 – 246.

[31] S. Kajita, Kanehiro, Kaneko, Fujiwara, Yokoi, and Hirukawa. A realtime pattern generator for biped walking. In *Proc. of the 2002 ICRA*, pages 31–27, 2002.

[32] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, , and H. Hirukawa. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *Proc. 2003 IROS*, page 16441650.

[33] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626, 2003.

[34] S. Kajita, Matsumoto, and Saigo. Real-time 3d walking pattern generation for a

biped robot with telescopic legs. In *Proc. of the 2001 ICRA*, pages 2299–2308, 2001.

[35] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 4489, 2010.

[36] S. Kajita, M. Morisawa, R. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, page 4489, 2010.

[37] S. Kajita and K. Tani. Study of dynamic locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1405–1411, 1991.

[38] S. Kajita and K. Tani. Experimental study of biped dynamic walking. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automations*, volume 3, pages 2885–2891, 1995.

[39] S. Kajita, T. Yamaura, and A. Kobayaashi. Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation*, 8:431–437, 1992.

[40] S. Kajita, T. Yamaura, and A. Kobayashi. Dynamic walking control of a biped

robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation*, 8:431 – 438, 1992.

[41] T. Kinugasa, T. Akiyama, M. Idris, K. Yoshida, and M. Iribe. Experimental analysis of 3d passive dynamic walking: Body's shape, com and stability. In *SICE Annual Conference 2010*, Oct 2010.

[42] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1146, 2003.

[43] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–92, 1990.

[44] T. McGeer. Passive walking with knees. In *Proceeding of the IEEE Conference on Robotics and Automation*, pages 1640–1645, 1990.

[45] T. W. Miller. Real time neural network control of a biped walking robot. *IEEE Control Systems Megazine*, pages 41–48, 1994.

[46] K. Mitobe, N. Mori, K. Aida, and Y. Nasu. Nonlinear feedback control of a biped walking robot. pages 2865 – 2870, 1995.

[47] H. Miura and I. Shimoyama. Dynamic walking of a biped. *Intl. journal of robotics research*, pages 60–74, 1984.

[48] J.-S. Moon and M. Spong. Bifurcations and chaos in passive walking of a compass-gait biped with asymmetries. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, page 1721, 2010.

[49] T. Mori, Y. Nakamura, M. Sato, and S. Ishii. Reinforcement learning for a cpg-driven biped robot. 2004.

[50] J. Morimoto, G. Cheng, C. Atkeson, and G. Zeglin. A simple reinforcement learning algorithm for biped walking. In *In Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 3030 – 3035. New Orleans, LA, April 2004.

[51] R. Murray and J. Hauser. *A case study in approximate linearization: The acrobat example*. University of California, Berkeley, College of Engineering, Berkeley, California, 1991.

[52] 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*, 47:79–91, 2004.

[53] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S.Schaal, and M. Kawato. A framework for learning biped locomotion with dynamical movement primitives. *Humanoids*, pages 925–939, 2004.

[54] Ogura, Shimomura, Kondo, Morishima, Okubo, Momoki, H. ok Lim, and Takanishi. Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3976 – 3981, 2006.

[55] D. Owaki, M. Koyama, S. Yamaguchi, S. Kubo, and A. Ishiguro. A two-

dimensional passive dynamic running biped with knees. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, page 5237, 2010.

[56] J. Park and K. Kim. Biped walking robot using gravity-compensated inverted pendulum mode and computed torque control. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, volume 4, pages 3528–3533.

[57] J. Park and Y. Rhee. Zmp trajectory generation for reduced trunk motions of biped robots. In *Proceedings of the 1998 IEEE International Conference on Intelligent Robots and Systems*, pages 90–96, 1998.

[58] J. H. Park and S. Lee. Generation of optimal trajectory for biped robots with knees stretched. In *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008, year = 2008, pages = 166,*.

[59] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: Biomechanics and control. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, page 2405, 2004.

[60] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 200 – 207, 2006.

[61] J. Pratt, C. M. Chew, Dilworth, and G. Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *Intl. journal of robotics research*, pages 129–143, 2002.

[62] J. Pratt and R. Tedrake. Velocity-based stability margins for fast bipedal walking. *Lecture Notes in Control and Information Sciences*, 340:299–324, 2006.

[63] J. Rebula, F. Canas, J. Pratt, and A. Goswami. Learning capture points for bipedal push recovery. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, page 1774, 2008.

[64] T. Reil and C. Massey. Biologically inspired control of physically simulated bipeds. *Theory Biosic*, 120:327 – 339, 2002.

[65] L. Righetti and A. J. Ijspeert. Design methodologies for central pattern generators:an application to crawling humanoids. In *Robotics: Science and Systems*, 2006.

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

[67] J. Rose and J. G.Gamble. *Human Walking*. LIPPINCOTT WILLIAMS and WILKINS, 2006.

[68] M. E. Rosheim. *Robot evolution: the development of anthrobotics*. Wiley-IEEE, 1994.

[69] S.Grillner and P. Zangger. How detail is the central pattern generation for locomotion. *Brain Research*, 88:367–371, 1975.

[70] C. Shih, Y. Zhu, and W. Gruver. Optimization of the biped robot trajectory. In

Proceedings of the IEEE International Conference on Robotics and Automation, 1991.

[71] C.-L. Shih, J. Grizzle, and C. Chevallereau. Asymptotically stable walking of a simple underactuated 3d bipedal robot. In *33rd Annual Conference of the IEEE Industrial Electronics Society, 2007 (IECON 2007)*, pages 2766 – 2771, 2007.

[72] A. Shimada and N. Hatakeyama. High-speed motion control of inverted pendulum robots. In *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, page 307, 2006.

[73] Z. Song and G. Zhao. Ga-based optimization of biped robot gait control of cpg model. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pages V3–392, 2010.

[74] M. Spong. The swing up control problem for the acrobot. *IEEE Control Megazine*, pages 49–55, 1995.

[75] M. Spong and R. L. R. Mahony. An almost linear biped. In *Proceedings of the 39th IEEE Conference on Decision and Control*, page 4803, 2000.

[76] S. H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley Publishing Company.

[77] T. Suzuki and K. Ohnishi. Trajectory planning of biped robot with two kinds of inverted pendulums. In *IEEE International Power Electronics and Motion Control Conference*, 2006.

[78] G. Taga. A model of the neuro-musculo-skeletal system for human locomotion i. emergence of basic gait. *Biological Cybernetics*, 73:97–111, 1995.

[79] G. Taga, Y. Yamaguchi, and H. Shimizu. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991.

[80] A. Takanishi, M. Tochizawa, H. Karaki, and I. Kato. Dynamic biped walking stabilised with optimal trunk and waist motion. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, The Autonomous Mobile Robots and Its Application, 1989.

[81] M. Vukobratovic and B. Borovac. Zero-moment point - thirty-five years of its life. *International Journal of Humanoid Robotics*, 1:157–173, 2004.

[82] M. Vukobratovic and B. Borovac. Zero moment point-thirty-five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–73, 2004.

[83] M. Vukobratovic, A. Frank, and D. Juricic. On the stability of biped locomotion. *IEEE Transactions on Biomedical Engineering*, 1970.

[84] M. Vukobratovic and J. Stepanenko. On the stability of anthropomorphic systems. *Mathematical Biosciences*, 15:1–37, 1970.

[85] Webots. Commercial mobile robot simulation software (www.cyberbotics.com).

[86] L. Yang, C. M. Chew, and A. N. Poo. Real-time bipedal walking adjustment modes using truncated fourier series formulation. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, page 379, 2007.

[87] L. Yang, C. M. Chew, A. N. Poo, and T. Zielinska. Adjustable bipedal gait generation using genetic algorithm optimized fourier series formulation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2006*, pages 4435–444, 2006.

[88] 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*, 25(5):549–565, 2007.

Author's Publications

[1] Van-Huan Dau, Chee-Meng Chew and Aun-Neow Poo, Proposal of Augmented Linear Inverted Pendulum Model for Bipedal Gait Planning, *The 2010 IEEE/RSI International Conference on Intelligent Robots and Systems (IROS2010)*, Oct 18-22, 2010, Taipei, Taiwan.

[2] Adiwahono A.H., Chee-Meng Chew, Weiwei Huang, Van-Huan Dau, Humanoid robot push recovery through walking phase modification, *The 2010 IEEE Conference on Robotics Automation and Mechatronics (RAM)*, 28-30 June 2010 , Singapore.

[3] Van-Huan Dau, Chee-Meng Chew, Aun-Neow Poo, Planning bipedal walking gait using Augmented Linear Inverted Pendulum model, *The 2010 IEEE Conference on Robotics Automation and Mechatronics (RAM)*, 28-30 June 2010 , Singapore.

[4] Van-Huan Dau, Chee-Meng Chew and Aun-Neow Poo, Achieving Energy-efficient Bipedal Walking Trajectory Through GA-Based Optimization of Key Parameters, *International Journal of Humanoid Robotics*, **Vol. 6**, Issue 4, pp. 609-629, 2009.

[5] Van-Huan Dau, Chee-Meng Chew and Aun-Neow Poo, Optimized Joint-Torques

Trajectory Planning for Bipedal Walking Robots, *3rd CIS-RAM IEEE International Conference*, Sept 21-24, 2008, Chengdu, China.

[6] Van-Huan Dau, Chee-Meng Chew and Aun-Neow Poo, Optimal trajectory generation for bipedal robots, *7th IEEE-RAS International Conference on Humanoid Robots*, Nov 29 - Dec 1, 2007, Pittsburgh, Pennsylvania, USA.

[7] Van-Huan Dau, Chee-Meng Chew and Aun-Neow Poo, Using Virtual Model Control And Genetic Algorithm To Obtain Stable Bipedal Walking Gait Through Optimizing The Ankle Torque, *10th International Conference on Climbing and Walking Robots*, 16 -18 July, 2007, Singapore (nominated for best paper award).

APPENDIX

8.3 The Optimal Values of T and K_v

		$\dot{\mathbf{x}}_\mathbf{i}$				
		0.3	**0.35**	**0.4**	**0.45**	**0.5**
$\mathbf{x_i}$	**-0.17**	1.575	1.491	1.73	1.751	1.442
	-0.14	1.547	1.596	1.568	1.702	1.751
	-0.11	1.322	1.685	1.87	1.418	1.24
	-0.08	1.428	1.407	1.168	1.309	0.9513
	-0.05	1.203	1.107	0.9216	0.7586	0.6548
	-0.02	0.6845	0.6548	0.5733	0.5511	0.5214
	0.00	0.5926	0.5084	0.4663	0.4382	0.4312

Table 8.1: Optimal Values of T (part 1)

		$\dot{x}_i$							
		0.5	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**
x_i	**-0.26**	1.295	1.175	1.151	1.161	1.091	0.9155	0.8383	0.6839
	-0.23	1.277	1.248	1.181	1.126	0.9717	0.7751	0.6628	0.5856
	-0.2	1.144	1.281	1.112	0.9015	0.775	0.6488	0.5645	0.5154
	-0.17	1.137	1.21	0.922	0.747	0.6347	0.5505	0.5014	0.4593
	-0.155	1.166	1.129	0.8105	0.6845	0.5807	0.5214	0.4695	0.4399
	-0.14	1.181	0.8875	0.719	0.6067	0.5365	0.4873	0.4452	0.4101
	-0.125	0.9958	0.7956	0.6548	0.5585	0.4992	0.4547	0.4176	0.388
	-0.11	0.8549	0.8875	0.5996	0.5154	0.4663	0.4242	0.3961	0.368
	-0.095	0.7734	0.6178	0.5436	0.4769	0.4325	0.3954	0.3732	0.3509
	-0.08	0.6628	0.5645	0.4944	0.4452	0.4028	0.375	0.3469	0.3259
	-0.065	0.6252	0.5214	0.4547	0.4102	0.3732	0.3509	0.3287	0.3065
	-0.05	0.5365	0.4663	0.4171	0.382	0.354	0.3259	0.3048	0.2908
	-0.02	0.4522	0.3891	0.354	0.3259	0.3048	0.2842	0.2694	0.2546
	-0.00	0.382	0.3399	0.3189	0.2908	0.2697	0.2546	0.2546	0.2398

Table 8.2: Optimal Values of T (part 2)

		$\dot{x}_i$				
		1.2	**1.25**	**1.3**	**1.35**	**1.4**
x_i	**-0.26**	0.5214	0.499	0.4992	0.4695	0.462
	-0.23	0.4844	0.4547	0.4473	0.4325	0.4176
	-0.2	0.4251	0.4102	0.4102	0.3954	0.3806
	-0.17	0.388	0.3806	0.3732	0.3584	0.3509
	-0.155	0.3732	0.3584	0.3509	0.3435	0.3361
	-0.14	0.3509	0.3435	0.3361	0.3287	0.3213
	-0.125	0.3287	0.3287	0.3213	0.3065	0.3065
	-0.11	0.3139	0.3065	0.2991	0.2916	0.2842
	-0.095	0.2991	0.2916	0.2842	0.2768	0.2768
	-0.08	0.2842	0.2768	0.2694	0.2694	0.262
	-0.065	0.2694	0.262	0.2546	0.2546	0.2472
	-0.05	0.2546	0.2472	0.2398	0.2472	0.2472
	-0.02	0.2472	0.2324	0.2324	0.2398	0.2249
	0.00	0.2398	0.2249	0.2249	0.2249	0.2324

Table 8.3: Optimal Values of T (part 3)

		$\dot{\mathbf{x}}_\mathbf{i}$				
		0.3	**0.35**	**0.4**	**0.45**	**0.5**
$\mathbf{x_i}$	**-0.17**	1.443	1.128	1.022	0.7094	0.3895
	-0.14	1.129	0.8486	0.5043	0.326	-0.147
	-0.11	0.6825	0.2527	-0.2357	-0.58	-0.9585
	-0.08	-0.2161	-0.8	-1.259	-1.965	-2.045
	-0.05	-2.07	-2.807	-3.12	-3.105	-3.11
	-0.02	-3.435	-4.927	-4.828	-4.963	-5.011
	0.00	-8.238	-7.879	-7.659	-7.425	-7.168

Table 8.4: Optimal Values of K_v (part 1)

		$\dot{\mathbf{x}}_\mathbf{i}$							
		0.5	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**
$\mathbf{x_i}$	**-0.26**	1.999	1.672	1.44	1.21	0.853	0.443	0.0794	-0.358
	-0.23	1.891	1.686	1.266	0.8584	0.394	0.038	-0.372	-0.721
	-0.2	1.672	1.486	0.929	0.433	-0.01	-0.418	-0.775	-1.112
	-0.17	1.486	0.9683	0.36	-0.096	-0.506	-0.863	-1.22	-1.545
	-0.155	1.313	0.514	0.0744	-0.389	-0.751	-1.137	-1.454	-1.813
	-0.14	0.971	0.292	-0.199	-0.626	-1.022	-1.4	-1.735	-2.053
	-0.125	0.565	-0.079	-0.548	-0.927	-1.132	-1.686	-2.026	-2.358
	-0.11	0.1697	-1.056	-0.907	-1.259	-1.657	-2	-2.387	-2.729
	-0.095	-0.333	-0.76	-1.271	-1.635	-2.02	-2.363	-2.79	-3.154
	-0.08	-0.7045	-1.242	-1.684	-2.109	-2.465	-2.885	-3.247	-3.657
	-0.065	-1.459	-1.842	-2.226	-2.636	-2.965	-3.421	-3.846	-4.256
	-0.05	-1.967	-2.441	-2.88	-3.325	-3.73	-4.161	-4.592	-4.974
	-0.02	-4.333	-4.563	-4.96	-5.311	-5.733	-6.184	-6.56	-7.039
	-0.00	-7.187	-7.216	-7.245	-7.678	-8.114	-8.47	-8.123	-8.651

Table 8.5: Optimal Values of K_v (part 2)

		$\dot{\mathbf{x}}_\mathbf{i}$				
		1.2	**1.25**	**1.3**	**1.35**	**1.4**
$\mathbf{x}_\mathbf{i}$	**-0.26**	0.5458	0.4103	0.1551	0.0989	-0.015
	-0.23	0.1551	0.095	-0.0354	-0.1905	-0.3578
	-0.2	-0.0696	-0.2161	-0.3822	-0.5348	-0.7143
	-0.17	-0.4359	-0.6227	-0.8217	-0.9707	-1.149
	-0.155	-0.6996	-0.857	-1.051	-1.242	-1.437
	-0.14	-0.9536	-1.158	-1.364	-1.56	-1.75
	-0.125	-1.227	-1.524	-1.725	-1.842	-2.111
	-0.11	-1.634	-1.85	-2.062	-2.26	-2.448
	-0.095	-2.136	-2.359	-2.565	-2.758	-3.029
	-0.08	-2.766	-2.982	-3.195	-3.414	-3.635
	-0.065	-3.474	-3.718	-3.972	-4.136	-4.392
	-0.05	-4.358	-4.619	-4.885	-4.875	-4.993
	-0.02	-6.057	-6.615	-6.634	-6.659	-6.99
	0.00	-7.484	-8.161	-8.07	-7.985	-7.523

Table 8.6: Optimal Values of K_v (part 3)

8.4 Function Estimation of T and K_v

8.4.1 Function estimation of the step time T

• if $\dot{\mathbf{x}}_\mathbf{i} \leq 0.5$ (part 1)

$$T^1(x_i, \dot{x}_i) = \sum_{i=-2}^{5} \sum_{j=-2}^{3} C_{i,j}^{T1} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.1}$$

where

$$C_{i,j}^{T1} = \begin{pmatrix} 1.5750 & 1.4910 & 1.7300 & 1.7510 & 1.4420 \\ 1.5470 & 1.5960 & 1.5680 & 1.7020 & 1.7510 \\ 1.3220 & 1.6850 & 1.8700 & 1.4180 & 1.2400 \\ 1.4280 & 1.4070 & 1.1680 & 1.3090 & 0.9513 \\ 1.2051 & 1.1088 & 0.9230 & 0.7594 & 0.6553 \\ 0.6965 & 0.6739 & 0.5873 & 0.5659 & 0.5332 \\ 0.5926 & 0.5084 & 0.4663 & 0.4382 & 0.4312 \end{pmatrix} \tag{8.2}$$

The plot of T^1 is shown in Fig. 8.1.

• if $0.5 \leq \dot{\mathbf{x}}_\mathbf{i} \leq 1.2$ (part 2)

$$T^2(x_i, \dot{x}_i) = \sum_{i=-4}^{10} \sum_{j=-4}^{4} C_{i,j}^{T2} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.3}$$

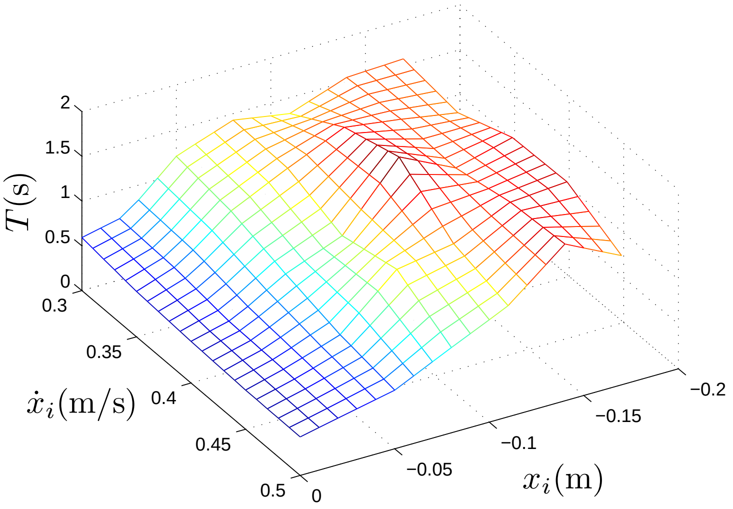


Figure 8.1: *T as a function of x_i and $\dot{x}_i$ (part 1)*

where

$$C^{T2}_{i,j} = \begin{pmatrix}
1.2950 & 1.1772 & 1.1303 & 1.1764 & 1.1208 & 0.7946 & 0.8762 & 0.6839 \\
1.4117 & 1.1231 & 1.1062 & 1.3400 & 1.0895 & 0.6720 & 0.7127 & 0.6035 \\
1.1107 & 1.5684 & 1.2911 & 0.9656 & 0.8339 & 0.6710 & 0.5124 & 0.5550 \\
1.1451 & 1.2453 & 1.1310 & 0.7338 & 0.6906 & 0.4870 & 0.5554 & 0.4641 \\
1.1265 & 1.6440 & 0.7333 & 0.7497 & 0.5602 & 0.5333 & 0.4307 & 0.4521 \\
1.2534 & 0.8779 & 0.7691 & 0.5878 & 0.5377 & 0.4679 & 0.4386 & 0.4088 \\
0.9769 & 0.6810 & 0.7151 & 0.5495 & 0.4961 & 0.4389 & 0.4003 & 0.3888 \\
0.8409 & 1.4399 & 0.4870 & 0.5327 & 0.4609 & 0.4081 & 0.3905 & 0.3664 \\
0.7896 & 0.4037 & 0.6178 & 0.4600 & 0.4337 & 0.3696 & 0.3768 & 0.3535 \\
0.6272 & 0.6638 & 0.4803 & 0.4516 & 0.3943 & 0.3736 & 0.3223 & 0.3220 \\
0.6477 & 0.4962 & 0.4672 & 0.3975 & 0.3581 & 0.3398 & 0.3232 & 0.3022 \\
0.3755 & 0.4656 & 0.3724 & 0.3600 & 0.3518 & 0.2757 & 0.2638 & 0.2764 \\
0.5011 & 0.3656 & 0.3612 & 0.3088 & 0.2846 & 0.2775 & 0.2646 & 0.2428 \\
0.3820 & 0.3387 & 0.3541 & 0.2887 & 0.2694 & 0.2437 & 0.2636 & 0.2398
\end{pmatrix} \quad (8.4)$$

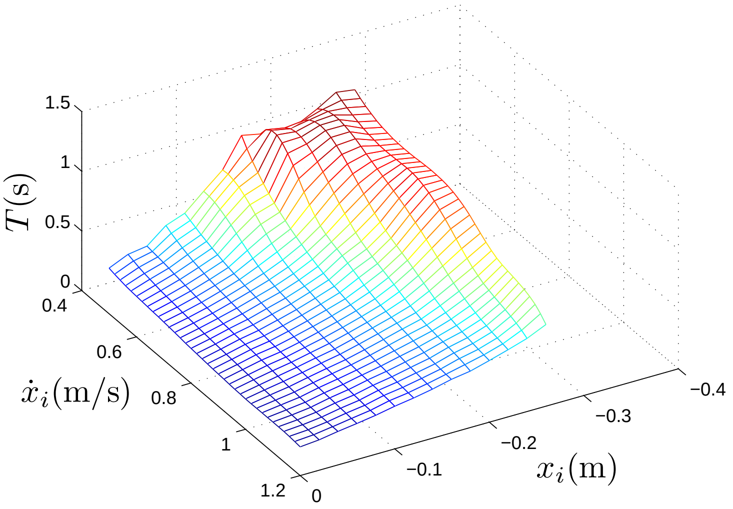


Figure 8.2: *T as a function of x_i and $\dot{x}_i$ (part 2)*

The plot of T^2 is shown in Fig. 8.2.

- **if $\dot{\mathbf{x}}_{\mathbf{i}} \geq 1.2$ (part 3)**

$$T^3(x_i,\dot{x}_i) = \sum_{i=-4}^{10}\sum_{j=-4}^{1} C_{i,j}^{T3} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.5}$$

where

$$C_{i,j}^{T3} = \begin{pmatrix} 0.5214 & 0.4815 & 0.5366 & 0.4422 & 0.4620 \\ 0.5175 & 0.4908 & 0.4601 & 0.4406 & 0.4331 \\ 0.4443 & 0.3900 & 0.4437 & 0.4076 & 0.3941 \\ 0.3941 & 0.4016 & 0.4070 & 0.3545 & 0.3604 \\ 0.3807 & 0.3592 & 0.3555 & 0.3415 & 0.3405 \\ 0.3527 & 0.3479 & 0.3314 & 0.3357 & 0.3211 \\ 0.3269 & 0.3317 & 0.3396 & 0.2860 & 0.3103 \\ 0.3144 & 0.3073 & 0.2912 & 0.2960 & 0.2788 \\ 0.2988 & 0.2926 & 0.2912 & 0.2639 & 0.2800 \\ 0.2832 & 0.2827 & 0.2540 & 0.2819 & 0.2607 \\ 0.2661 & 0.2624 & 0.2478 & 0.2478 & 0.2397 \\ 0.2371 & 0.2454 & 0.1870 & 0.2693 & 0.2604 \\ 0.2536 & 0.2254 & 0.2416 & 0.2472 & 0.2015 \\ 0.2398 & 0.2203 & 0.2286 & 0.2220 & 0.2324 \end{pmatrix} \tag{8.6}$$

The plot of T^3 is shown in Fig. 8.3.

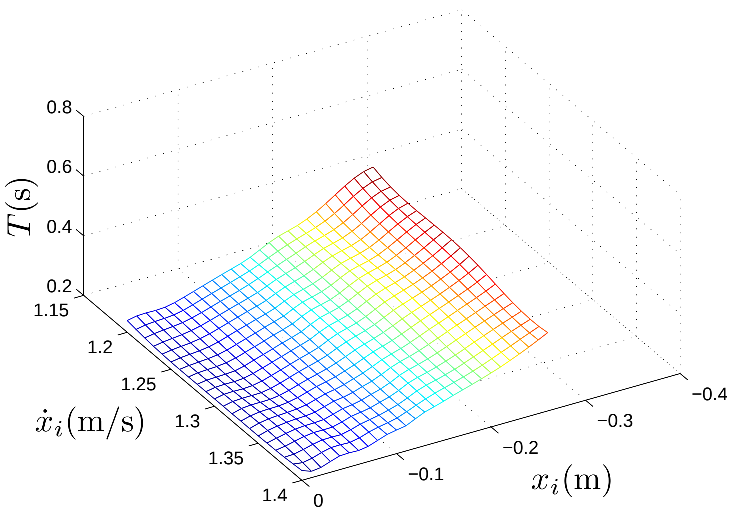


Figure 8.3: *T as a function of x_i and $\dot{x}_i$ (part 3)*

8.4.2 Function estimation of the parameter K_v

• if $\dot{x}_i \le 0.5$ (part 1)

$$k_v^1(x_i,\dot{x}_i) = \sum_{i=-4}^{3}\sum_{j=-4}^{1} C_{i,j}^{kv1} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.7}$$

where

$$C_{i,j}^{kv1} = \begin{pmatrix} 1.4430 & 1.0368 & 1.2622 & 0.5267 & 0.3895 \\ 1.3133 & 1.1899 & 0.8077 & 0.5280 & 0.0895 \\ 0.8510 & 1.0693 & -0.8478 & 0.6231 & -0.4467 \\ 0.2275 & -0.6649 & -0.0281 & -2.9834 & -2.0847 \\ -3.2525 & -2.9608 & -5.1313 & -2.3995 & -3.1378 \\ -1.7519 & -5.6874 & -2.8141 & -5.5356 & -5.0012 \\ -8.2380 & -7.9103 & -7.6923 & -7.3410 & -7.1680 \end{pmatrix} \tag{8.8}$$

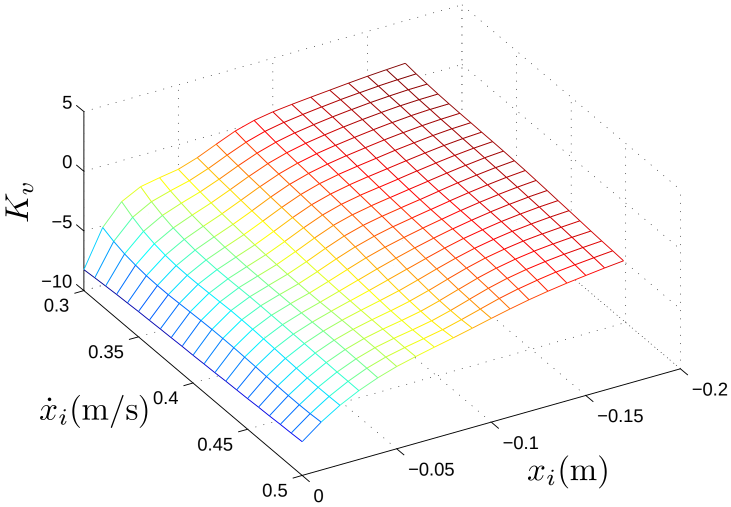


Figure 8.4: *Kv as a function of x_i and $\dot{x}_i$ (part 1)*

The plot of k_v^1 is shown in Fig. 8.4.

- **if $0.5 \leq \dot{x}_i \leq 1.2$ (part 2)**

$$k_v^2(x_i, \dot{x}_i) = \sum_{i=-4}^{10} \sum_{j=-4}^{4} C_{i,j}^{kv2} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.9}$$

where

$$C_{i,j}^{kv2} = \begin{pmatrix} 1.9990 & 1.7435 & 1.5032 & 1.2486 & 0.8550 & 0.2705 & -0.0118 & -0.3578 \\ 2.0100 & 2.2193 & 1.2779 & 1.1849 & 0.2681 & 0.4404 & -0.6834 & -0.5389 \\ 1.7572 & 1.4930 & 1.6232 & 0.5195 & 0.4543 & -0.6866 & -0.4227 & -0.9672 \\ 1.5435 & 2.0553 & 0.4908 & 0.2639 & -0.5500 & -0.5894 & -1.3931 & -1.3294 \\ 1.4309 & 0.4783 & 0.4180 & -0.4467 & -0.5303 & -1.3696 & -1.3495 & -1.7712 \\ 1.0133 & 0.3578 & -0.0134 & -0.5293 & -1.2105 & -1.3779 & -1.9175 & -2.0170 \\ 0.5518 & 0.6820 & -0.4642 & -1.0418 & -0.7473 & -2.2136 & -1.9010 & -2.3420 \\ 0.2210 & -2.6818 & -0.3512 & -1.2678 & -1.8218 & -2.0004 & -2.5979 & -2.7214 \\ -0.4190 & 0.2780 & -1.3780 & -1.5584 & -2.0136 & -2.3857 & -3.0350 & -3.1462 \\ -0.5847 & -1.4138 & -1.4227 & -2.1828 & -2.4775 & -3.1706 & -3.2698 & -3.6750 \\ -1.8033 & -1.7555 & -2.2107 & -2.7555 & -2.9598 & -3.6978 & -4.1484 & -4.3861 \\ -2.0515 & -2.9767 & -3.2893 & -4.0590 & -4.7403 & -5.1116 & -5.5151 & -5.6986 \\ -5.0221 & -4.4960 & -5.3927 & -5.4855 & -5.8505 & -6.6003 & -7.2148 & -7.5474 \\ -7.1870 & -7.3592 & -6.9371 & -7.7042 & -8.0834 & -8.8869 & -7.5241 & -8.6510 \end{pmatrix} \tag{8.10}$$

The plot of k_v^2 is shown in Fig. 8.5.

• if $\dot{\mathbf{x}}_\mathbf{i} \geq \mathbf{1.2}$ (part 3)

$$k_v^3(x_i, \dot{x}_i) = \sum_{i=-4}^{10} \sum_{j=-4}^{1} C_{i,j}^{kv3} N_{i,k+1}(\dot{x}_i) M_{j,l+1}(x_i) \tag{8.11}$$

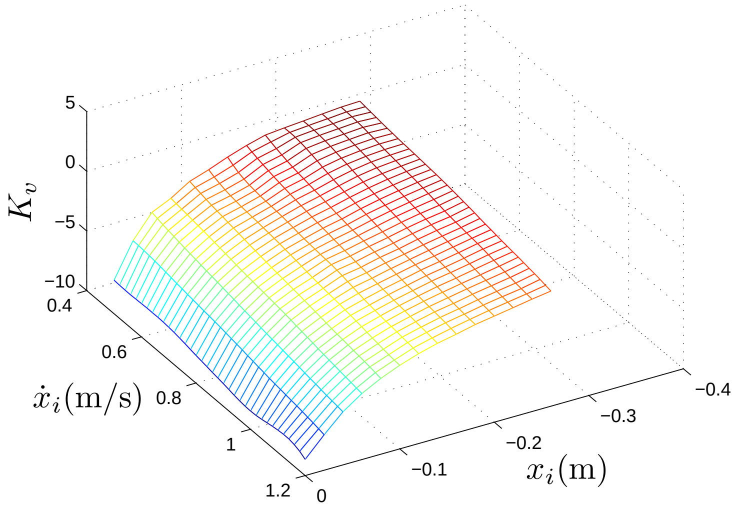


Figure 8.5: *Kv as a function of x_i and $\dot{x}_i$ (part 2)*

where

$$C_{i,j}^{kv3} = \begin{pmatrix} 0.5458 & 0.5982 & -0.0735 & 0.1692 & -0.0150 \\ 0.1905 & 0.2160 & 0.1208 & -0.1547 & -0.2113 \\ 0.0680 & 0.0283 & -0.0834 & -0.3782 & -0.5528 \\ -0.2412 & -0.3750 & -0.7948 & -0.8218 & -0.9402 \\ -0.6373 & -0.7032 & -0.9831 & -1.1737 & -1.3668 \\ -0.9402 & -1.0403 & -1.2855 & -1.6833 & -1.7029 \\ -1.1895 & -1.4676 & -1.8977 & -1.7006 & -2.1368 \\ -1.6251 & -1.7286 & -1.9834 & -2.3788 & -2.3723 \\ -2.1139 & -2.2773 & -2.6133 & -2.7279 & -3.0641 \\ -2.8072 & -2.9537 & -3.1596 & -3.5698 & -3.6158 \\ -3.5997 & -3.7284 & -4.2739 & -4.2844 & -4.6592 \\ -5.3812 & -5.3040 & -6.4756 & -5.2916 & -5.3285 \\ -6.2243 & -7.3145 & -6.4659 & -7.1224 & -7.9580 \\ -7.4840 & -8.4276 & -7.8732 & -8.1061 & -7.5230 \end{pmatrix} \tag{8.12}$$

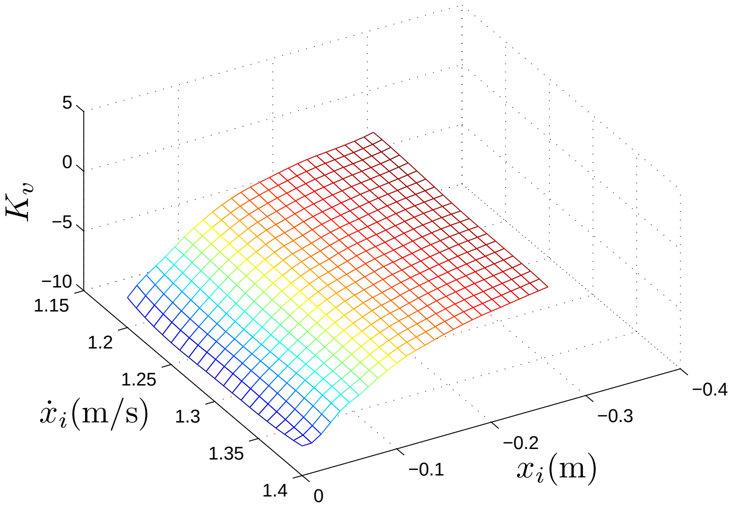


Figure 8.6: *Kv as a function of x_i and $\dot{x}_i$ (part 3)*

The plot of k_v^3 is shown in Fig. 8.6.