BEHA VIOURAL ANALYSIS OF BIPEDAL ROBOT

A dissertation submitted to the
Department of Electrical Engineering, University of Moratuwa
in partial fulfilment of the requirements for the
Degree of Master of Science

by
SAMAN MANJULA WELIHINDA

Supervised by: Prof. Lanka Udawatta

Department of Electrical Engineering
University of Moratuwa, Sri Lanka

2010

Abstract

In this thesis a 7 OoF bipedal robot has been simulated, and analyzed its behavior by varying torso angle to achieve dynamic stability while walking on sloping surfaces. The revolutionary dynamic stability criteria introduced by Prof. Miomir Vukobratovic in 1969 has been used throughout the thesis. The dynamically stable human walking on slopes are very complex and this thesis address this problem by starting from lower body and in the middle by adding a torso with the intention of gaining dynamic stability and finishes showing the effectiveness of the variation of torso angle.

The ZMP formula presented in the paper of Prof. Miomir Vukobratovic is involved with huge computations and due to that most of the researches in this field have chosen alternatives such as GA and NN to find the ZMP. Although it is convenient to use GA and NN to avoid mathematical calculation the accuracy of resultant ZMP is questionable. The original ZMP equation has been used extensively in all cases to calculate ZMP in this thesis and rather verifies the accuracy of this concept.

The simulations have been performed to verify the accuracy of the kinematics models that has been created before moving to the ZMP calculations. The effectiveness of the variation of torso angle on the dynamic stability of the bipedal robot has been analyzed by varying slope angle, step length and step time. It is indicated using real ZMP calculations the bipedal robot can maintain its dynamic stability while walking on slopes at any circumstances.

DECLARATION

The work submitted in this dissertation is the result of my own investigation, except where otherwise stated.

It has not already been accepted for any degree, and is also not being concurrently submitted for any other degree.

UOM Verified Signature

Saman Manjula Welihinda

Date -08-02-2010

I endorse the declaration by the candidate.

UOM Verified Signature

Prof. Lanka Udawatta

Contents

Acknowledgement

Thanks are due primarily to my supervisor, Professor Lanka Udawatta, for the contribution of his great insights, perspectives, correct-guidance and sense of humor.

In all, my sincere thanks go to the officers in the Post Graduate Office, Faculty of Engineering, and to the University of Moratuwa of Sri Lanka for the corporation extended to me in numerous ways for the clarification of matters related to my academic studies on time throughout the course.

Also my sincere gratitude extended to the staff who serves in the Department of Electrical Engineering office.

I also wish to thank my fellow engineers in office for their generous support given from the beginning to the end of the course.

I also extend my sincere thanks to Mr. Aruna Abeyratne for deriving the equations necessary for the simulations and calculations.

I should thank many individuals, friends and colleagues who have not been mentioned here personally in making this educational process a success.

Last but not least my gratitude goes to my wife for the love, moral support and understanding from start to end of this course.

List of Abbreviations

α	Slope Angle
DoF	Degrees of Freedom
g	Acceleration due to gravity
SL	Step length
T	Step Time
D-H	Denavit-Hartenberg notation
L_i	Length of link i
ZMP	Zero moment point
FRI	Foot rotation index
FZMP	Fictional zero moment point
L_t	Length of Torso
m_t	Mass of Torso
θ_t	Torso angle
GA	Genetic algorithm
NN	Neural network
COM	Centre Of Mass
NPCM	Normal Projection of the Centre of Mass
C_i	Cosine of joint angle i
S_i	Sine of joint angle i
$q_i(t)$	Joint angle at time t

List of Figures

List of Tables

Chapter 1

Introduction

In recent years there has been much interest simulated in bipedal robotics and legged locomotion in general. Part of the reason for this interest is the need for robots which can operate in human oriented environments. However, the biggest challenge is biped walking like that of a human being, which was improved over many centuries of evolution. Since approximately 1980, the research has been more focussed on biped walking.

1.1 Bipedal robot

The human beings and almost all on land living animals use legs for locomotion. However not many machines were built using legs for movement. The reasons therefore are the complex design and control. Nevertheless, the main advantage of waking machines is that in contrast to wheeled robots, they do not need a customized environment. They could be able to move in an environment that is only accessible by human beings. In theory not only walking but also running, jumping, climbing or even swimming could be implemented. In contrast, wheeled machines need a relative planar terrain and enough space to avoid obstacles. Bipeds use different support areas for carrying their weight and getting grip and are in the ideal case as fast and flexible as a human. Using this flexible support on the ground, a large adaptability is achieved. The legs can also be considered as an individual suspension system whereby the upper part of the body moves forward on another trajectory as the feet. Decoupling the legs from the rest of the body allows carrying payload smooth through a rough terrain. Both types of robots are designed for a specified environment: The wheeled robots are more efficient on a planar surface whereas walking machines have an advantage on all other terrains. The operational area of robots, especially with two legs, is the natural setting of humans. The human body has, because of his anatomy, an exceptionally maneuverability which is perfect exploited for his locomotion. Thus, he can adapt to a new environment with minimal effort.

1.2 Basics of bipedal walking

To understand the topic of the biped walking an overview of a human model will be shown. For the reason that most of the humanoid robots use the human body as paragon, it is suggestive to use the same terminology as for the human anatomy. There are three basic planes referred to as frontal (or coronal), sagittal and transversal as shown in Figure 1.1.

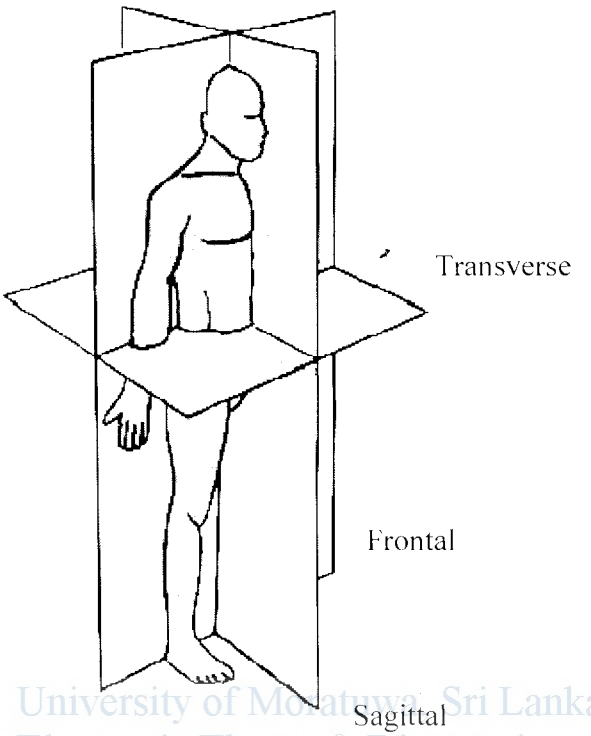


Figure 1.1: Anatomical planes

1.2.1 Gait Phases

Walking is a cyclic movement consisting of two main phases, which alternates on both legs and illustrated in Figure 1.2 and 1.3.

Double support phase is the phase during which both feet are in touch with ground. In single support phase only one foot is in touch with ground while other foot is in swinging.

During the double support phase (I), both feet are in contact with the ground. In this phase, the body has a stable position because of the wide support area on the ground. The system enters this state with the heel strike (IV) and exits it with the toe off (II) movement.

During the single support phase, only one foot is in contact with the floor. In this state, the centre of mass (COM) of the system rotates like an inverted pendulum above the contact point. Meanwhile the swing leg moves forward (III) to touch the ground again and enter the other phase.

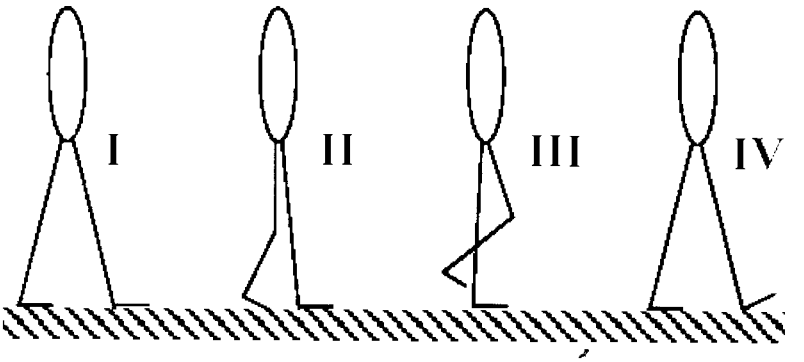


Figure 1.2: Leg positions during one half cycle

The Figure 1.2 show the four leg positions during one half cycle while walking. The complete walking cycle illustrated in Figure 1.3. The right leg and left leg is going through the same sequence alternately and this repeats while walking.

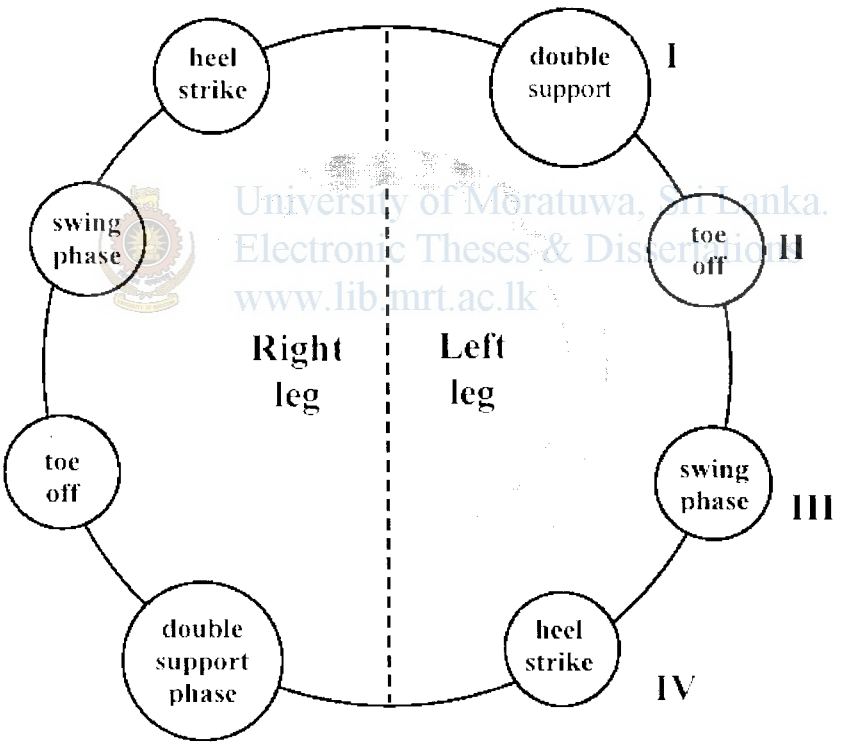


Figure 1.3: The cyclic phase rotation of bipedal walking

1.2.2 Static and Dynamic Walking

Walking can be divided into two main groups: walking with static balance and walking with dynamic balance. The Figure 1.4 shows the gait cycle of a bipedal robot.

During a static walk [1] the normal projection of the centre of mass (NPCM) always stays in between the boundaries defined by the feet. If both feet are on the ground, the NPCM has to be within the polygon determined by the outer corners of the biped feet. If only one foot is in contact with the ground, the NPCM has to be within the area of this foot. While the movement is slow enough, the system dynamics can be ignored. Static walking assumes that if the system's motion is stopped at any time, it will stay in a stable position indefinitely. However, the speed achieved using static walk is not that high and the efficiency is far away from the human walking speed.

In contrast, a dynamically or actively balanced walk [2] is not constrained in such a manner. The COM may leave the support area formed by the feet for periods of time. This allows the system to experience tipping moments, which give rise to horizontal acceleration. However, such periods of time are kept brief and strictly controlled so that the system does not become unstable. Thus one may think of a dynamically balanced system as one where small amounts of controlled instability are introduced in such a way as to maintain the overall equilibrium. Tipping moments in one direction are negated by tipping moments in the opposite direction.

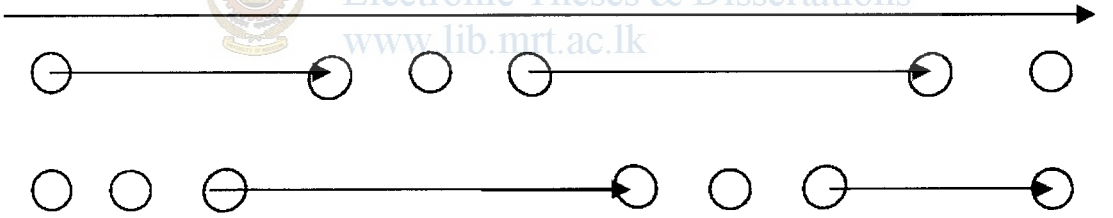


Figure 1.4: A bipedal robot's gait cycle; the red circles indicate the footprints; the shaded area is the supporting area; the leg movement is expressed by the arrows

When comparing the two methods of balance, it can be seen that the static method is highly restrictive and results in movement which is slow. Very rarely do animals and humans exhibit such behavior for this reason the velocity achievable is very low and the motion is not efficient. However, it can see that by removing the constraining nature of the rule for static balance that the mobility of the system is increased. This is due to the increased edibility of the movement of the legs and placement of the feet. The accelerating tipping moments can be used to achieve higher speeds, move all legs at once or to utilize footholds which are far apart.

Therefore it can be seen that in order for a bipedal robot to gain efficiency and speed, it will require dynamic balance.

1.2.3 Zero moment Point

Perhaps the most widely used classical approach to dynamic biped walking are those control systems based on the measurement of zero moment point (ZMP) [3][4]. The ZMP is a tool used to measure the dynamic stability of a walking system. In and of itself, the ZMP is not a complete control system. Rather, the ZMP measurement is used by a control system in the same way that the centre of mass projected onto the ground (GCoM) can be used for static walking systems. It is used in the offline generation of dynamically stable walking gaits, and online to predict if a system needs to take some kind of corrective action to prevent the loss of its dynamic stability.

Dynamic stability, as defined by researchers using ZMP walking strategies, has a specific meaning: In order for a biped walker to be dynamically stable at a particular in time, its feet must remain motionless with respect to the ground for the entire time that they support the robot's weight. If a robot begins to "tip over" the edge of its support foot, then it is no longer considered dynamically stable.

The ZMP is defined as the point within the supporting polygon of a dynamically stable structure at which a single ground reaction force acts in order to cause the sum of all moments of active forces on the robotic system to equal zero. The diagram below (Figure 1.5) shows a representation of a typical robot foot.

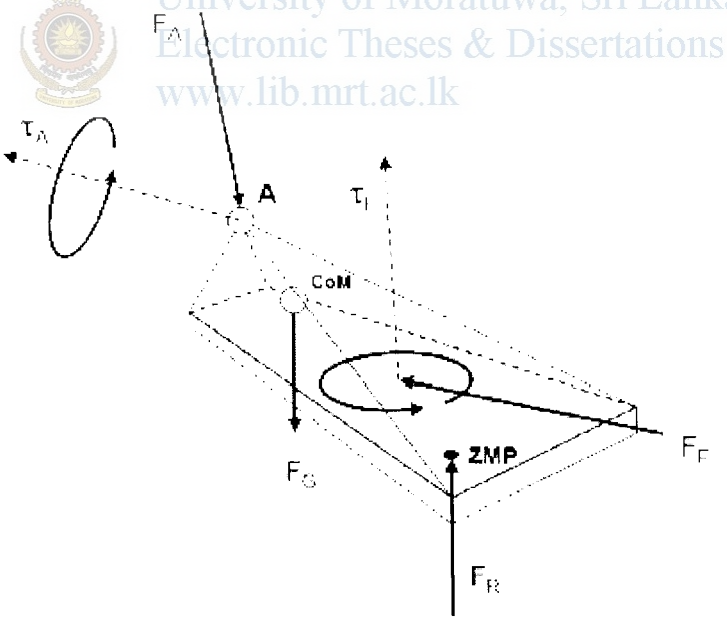


Figure 1.5: Forces and torque acting on a robot foot

Interaction between the foot and the rest of the robot's structure can be represented by a single force F_A and torque τ_A acting on the foot's ankle joint. Additional forces working on the system include:

- F_G, the gravitational force acting on the foot, through the foot's centre of mass
- F_F and t_F the frictional force and torque preventing the foot from sliding or twisting along the ground, and
- F_R, the ground reaction force preventing the foot from penetrating the ground surface.

The ZMP definition of a dynamically stable robot requires the foot does not roll or tilt with respect to the ground during gait execution. This means that to be stable, the resultant of all the pitch and roll torques acting on the foot must be zero. It is left to the ground reaction force F_R to cancel out any horizontal components of torque resulting from τ_A and horizontal torque contributions of all other forces. Since the ground reaction force is fixed in magnitude, the only way it can compensate for varying horizontal torque is to be applied at different points on the robot's supporting polygon. Note that it is possible for a robot's foot to slide while still satisfying the ZMP stability criteria.

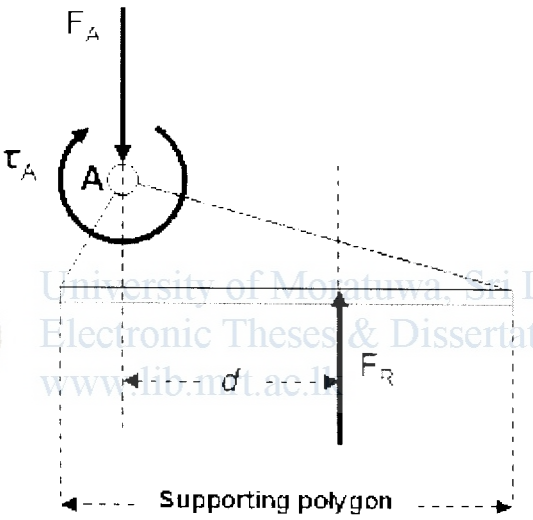


Figure 1.6: Simplified foot dynamics

In the example depicted by Figure 1.6 above, as the distance d increases, the reactive torque produced by F_R also increases, such that:

$$\tau_R = d * F_R \qquad (1.1)$$

If a point can be found within the supporting polygon such that the reactive torque τ_R applied at this point exactly cancels out all other torques acting on the foot, then this point is known as the zero moment point, and the robot is dynamically stable. In the example above, when d coincides with the ZMP, the reactive force F_R will cancel out all other torques, so if all other forces are known, the location of the ZMP can be calculated:

$$\tau + \tau_R = 0 \qquad (1.2)$$

- F_G, the gravitational force acting on the foot, through the foot's centre of mass
- F_F and t_F the frictional force and torque preventing the foot from sliding or twisting along the ground, and
- F_R, the ground reaction force preventing the foot from penetrating the ground surface.

The ZMP definition of a dynamically stable robot requires the foot does not roll or tilt with respect to the ground during gait execution. This means that to be stable, the resultant of all the pitch and roll torques acting on the foot must be zero. It is left to the ground reaction force F_R to cancel out any horizontal components of torque resulting from τ_A and horizontal torque contributions of all other forces. Since the ground reaction force is fixed in magnitude, the only way it can compensate for varying horizontal torque is to be applied at different points on the robot's supporting polygon. Note that it is possible for a robot's foot to slide while still satisfying the ZMP stability criteria.

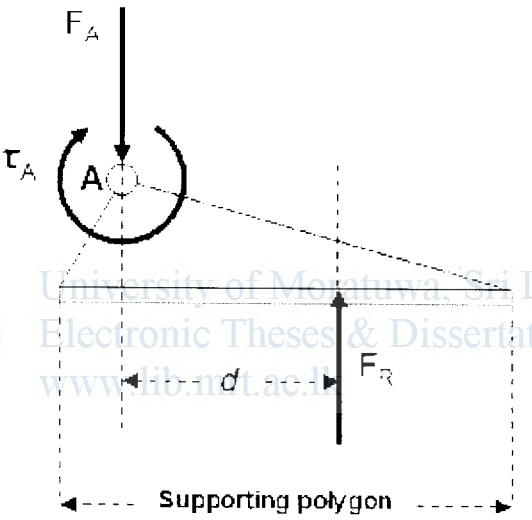


Figure 1.6: Simplified foot dynamics

In the example depicted by Figure 1.6 above, as the distance d increases, the reactive torque produced by F_R also increases, such that:

$$\tau_R = d^* F_R \qquad (1.1)$$

If a point can be found within the supporting polygon such that the reactive torque τ_R applied at this point exactly cancels out all other torques acting on the foot, then this point is known as the zero moment point, and the robot is dynamically stable. In the example above, when d coincides with the ZMP, the reactive force F_R will cancel out all other torques, so if all other forces are known, the location of the ZMP can be calculated:

$$\tau + \tau_R = 0 \qquad (1.2)$$

$$\tau_{ii} + d_{ZMP} * F_R = 0 \quad (1.3)$$

$$d_{ZMP} = - \tau_{ii} / F_R \quad (1.4)$$

As torque transmitted to the foot by the rest of the robot system increases, the ZMP must move closer to the edge of the foot in order to generate enough reactive torque to maintain dynamic stability. Once the ZMP reaches the foot's edge, the system is only marginally stable and any further increase in applied torque will cause the system to tip over. So in dynamically balanced systems, the ZMP has the same relationship to the robot's supporting polygon as the GCoM does in statically balanced systems. As the ZMP moves towards the border of the robot's supporting polygon, the robot's dynamic stability becomes more marginal.

By definition, the ZMP can never exist outside of the robots supporting polygon. If the robot is not dynamically stable, then there is no zero moment point. Sometimes it is useful to be able to compare the relative stability of various unstable states. To this end, researchers have extended the ZMP concept to include an imaginary point outside of the robot's supporting polygon, where the ground reaction force would have to act if it were to maintain dynamic stability. This imaginary force is calculated in the same way as the ZMP, with the constraint to remain within the supporting polygon removed. In order to highlight the difference between this imaginary point, and a real ZMP, the terms foot rotation index (FRI) or fictional zero moment point (FZMP) have been suggested [5]. Since it is impossible to apply a force to an object without contacting it, while it is outside the robot's supporting polygon, the FRI does not exist and so cannot be measured. While the FRI lies within the robot's supporting polygon, it is equivalent to ZMP and the robot is dynamically stable. Since the FRI is not always measurable, and mainly applies to systems that have already lost dynamic stability, it is not very useful for on-line ZMP based control systems. FRI/FZMP calculations are more usefully applied to off-line gait synthesis tasks.

1.3 Review and Research

There has been much research performed by both universities and private organizations in the area of bipedal walking in the past 40 years. It is not possible to outline all the developments that have been made in a project of this size. Hence brief overview of bipedal robots will be given. The currently best known examples of the state-of-the-art in humanoid robots are the Honda ASIMO [6] and the Sony QRIO [7] and both use ZMP based control approaches.

The Honda Company in Japan has presented its first two legged robot E0 in 1986.The E0 walked slowly in a straight line. After the introduction of E0 other two legged robots E1-E2-E3 as shown in Figure 1.7 has been developed from 1987-1991 to achieve fast walking. The E2 robot achieved fast walking at a speed of 1.2 Km/h on flat surfaces.

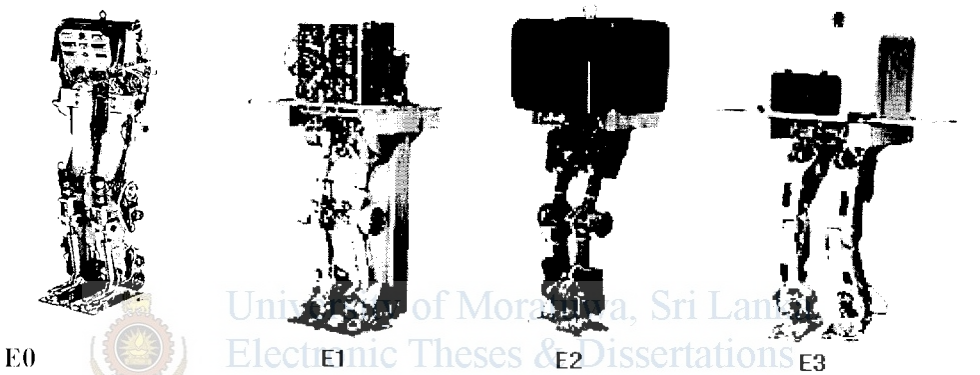


Figure 1.7: The bipedal robots E0-E3 introduced by Honda from 1986-1991

The robots E4-E5-E6 showing in Figure 1.8 presented between 1991-1993 establishing techniques for stable walking and developed three control techniques.

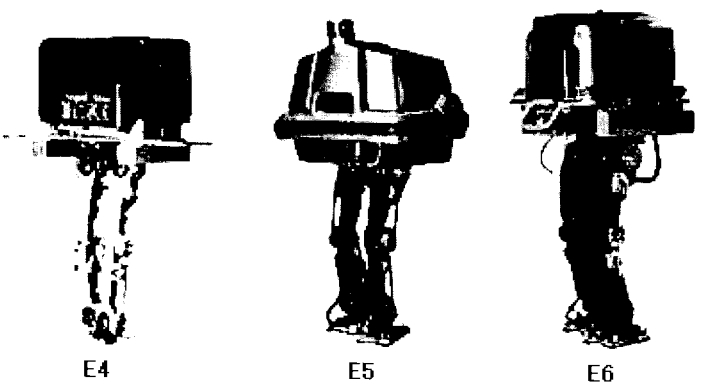


Figure 1.8: The bipedal robots E4-E6 introduced by Honda from 1991-1993

After the seven basic models the HONDA Company changed the research direction towards humanoid robot. The humanoid robots P1, P2, P3 showing in Figure 1.9 were

presented from 1993-1997. P1 is the first Honda prototype of a humanoid robot. This robot can turn electrical and computer switches, grab door knobs, pickup and carry things. The height of P1 is 1.915 m and 175 Kg. P2 is a self regulating battery-operated android which can walk independently, walk up and down the stairs, and push carts. Wireless techniques were used with a computer torso, motors, battery, wireless radio and other built-in devices. P2 is 1.82 m tall and 210Kg. P3 changes in component materials and a decentralized control system which resulted in decrease in height, 1.6m and weight 130 Kg. It has 16 joints in total. It has 30 degrees of freedom; 12 for legs, 14 for arms and 4 for hands. The maximum walking speed of P3 is 2Km/hr.

Figure 1.9: The humanoid robots P1-P3 presented by Honda from 1993-1997

ASIMO (Advanced Step in Innovative Mobility) was presented in 2001 to the world.(see Figure 1.10). With a weight of 43 Kg, he is much lighter than his predecessors are and that is one of the most important advantages.

This robot is the result of about 14 years of research and approximately 10 prototypes. ASIMO can walk continuously while changing directions, taking every step smooth and natural. He also has the ability to climb up and down a flight of stairs.

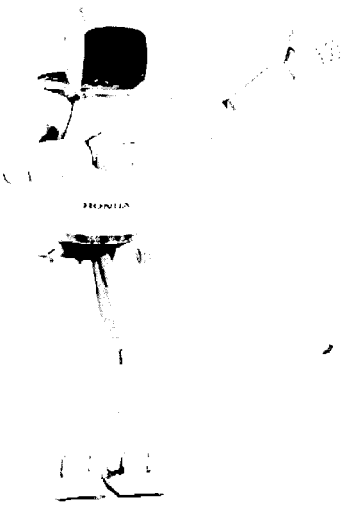


Figure 1.10: The humanoid robot ASIMO

In December 2002 Honda added intelligence technology to ASIMO which is capable of interpreting the postures and gestures of humans and moving independently in response. ASIMO's ability to interact with humans has advanced significantly; it can greet approaching people, follow them, and move in the direction they indicate, and even recognize their faces and address them by name. Further, utilizing network such as Internet, ASIMO can provide information while executing tasks such as reception duties.
Honda debuted a new ASIMO humanoid robot which features the ability to pursue key tasks in a real-life environment such as an office and an advanced level of physical capabilities. Compared to the previous model, the new ASIMO achieves the enhanced ability to act in sync with people. In addition, the running capability is dramatically improved, with ASIMO now capable of running at a speed of 6Km/h and of running in a circular pattern. New ASIMO weights 54Kg and total height is 130cm and normal walking speed is 2.7Km/h and consists overall 34 degrees of freedom.

QRIO ("**Q**uest for cu**RIO**sity", originally named Sony Dream Robot or SDR) in Figure 1.11 was a bipedal humanoid robot developed and marketed (but never sold) by Sony Corporation to follow up on the success of its AIBO toy. QRIO stood approximately 0.6 m tall and weighed 7.3 kg.

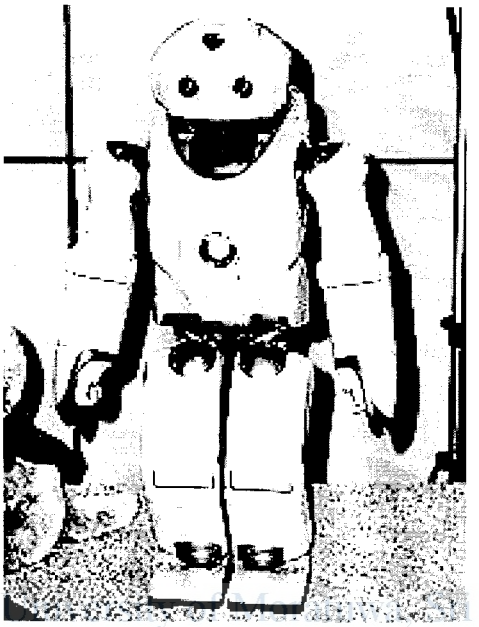


Figure 1.11: Humanoid robot "QRIO" introduced by Sony

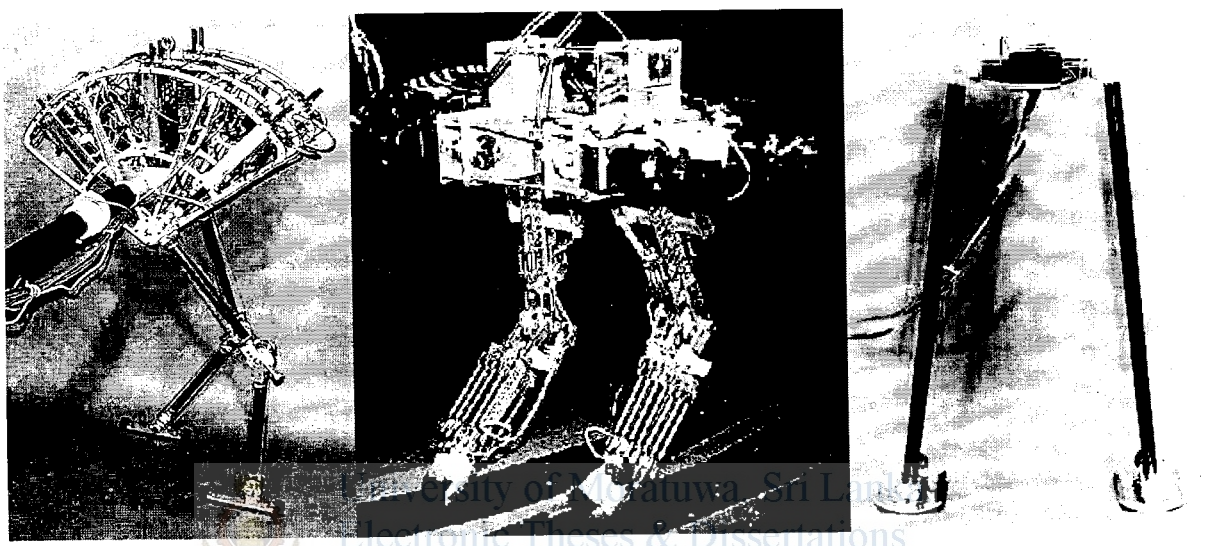
QRIO is capable of voice and face recognition, making it able to remember people as well as their likes and dislikes. QRIO can run at 23 cm/s, and is credited in Guinness World Records (2005 edition) as being the first bipedal robot capable of running (which it defines as moving while both legs are off the ground at the same time). The 4th generation QRIO's internal battery lasts about 1 hour. A special feature of QRIO is that it reacts to protect itself against an impact. In the event of falling, it gets back up by itself after checking in all directions.

The Massachusetts Institute of Technology (MIT) [8] was one of the first in developing a variety of walking, running or jumping machines. The MIT has come up with several robots as shown in Figure 1.12. The walking robot Geekbot used to study the smooth transfer of support from one foot to the other. The robot is designed for motion in the coronal plane only. The GeekBot consists of two curved feet, actuated ankles and hips, and a pelvis. The basic behavior is controlled rocking from one foot to the other. This research has carried out from 1994 to 1995.

Spring Turkey was designed and built by Peter Dilworth and Jerry Pratt in 1994. It has an actuated hip and knee on each leg. An unactuated boom constrains Spring Turkey's roll,

yaw, and lateral motion thereby reducing it to a two dimensional robot. Spring Turkey weighs in at approximately 22 lbs (10 kg) and stands 2 ft (60 cm) tall.

A new robot, Spring Flamingo, was started after the Spring Turkey. Spring Flamingo Weighs in approximately 30 lbs (13.5 kg) and stands 3 ft (90 cm) tall. The robot has an actuated hip, knee, and ankle on each leg. An unactuated boom constrains Spring Flamingo's roll, yaw, and lateral motion thereby reducing it to a planar robot.



(a) (b) (c)

Figure 1.12: MIT humanoid robots: (a) Spring Flamingo, (b) Spring Turkey, (c)GeekBot

Many other notable, successful bipeds with stiff legs and large feet exist. (e.g. HRP-2 [9], H6 [10], HOAP-2 [11], Johnnie [12], KHR-1 [13]) and some are even available commercially.

1.4 Motivation

Legged locomotion is the most convenient form of locomotion when compared to the other forms of locomotion due to the ability of negotiating complex terrains. Legged robots can be used for lot of practical applications such as emergency fast response [14], mobile monitoring , inspection and surveillance in hazardous or hostile environments [15,16], planetary exploration [17], and personal applications, to name a few.

Bipedal robots are especially interesting because humans are two-legged and places meant to be occupied by humans (such as homes and offices) are designed for two legged locomotion.

The HONDA Company has come up with the state of the art humanoid robot ASIMO after 14 years research with billions of dollars investment. But still there are lots of improvements are needed to make bipedal robots commercially viable. In the future the human labor will become so expensive and there will be lot of demand for the autonomous labor.

As mobility is a key element in developing autonomous vehicles, legged robots will be of increasing significance as there is a trend towards more capable autonomous robot continues. Still there are no practical applications of bipedal robots outside the laboratory apart from number of entertainment robots.

Although there were hundreds of researches carried out on bipedal robots still there is lot more to learn about bipedal walking. When going through the literature it is noted that there are very few researches are carried out on bipedal robots which can walk on sloping surfaces. Author also noted that the use of original ZMP equation presented by Prof. Miomir Vukobratovic so much rare and most of the time researches has used artificial intelligence techniques to test dynamic stability but this surely does not give exact result but approximated value.

Therefore due to the above reasons the author has motivated to study about a bipedal walking on sloping surfaces and testing of its dynamic stability during different circumstances by calculating its dynamic stability using original ZMP equation.

1.5 Research Objectives

The aim of this research is the simulation and analyzing of the bipedal robot walking on sloping surfaces and to verify the ability of maintaining dynamic stability by varying torso angle. To achieve this objective a bipedal robot consists of seven links having two legs and a torso has been modeled. The dynamic stability has been tested for the period of swing phase of the leg using predefined trajectories using the ZMP concept introduced by the Prof. Miomir Vukobratovic. The original ZMP equation has been used for the calculation of ZMP at each trajectory point.

1.6 Overview

The structure of this thesis is divided into seven chapters.Chapter 1 reviews past literature and the current state of research in bipedal robotics. In Chapter 2 the bipedal robot description and trajectory planning has been discussed.
The chapter 3 discusses the application of kinematics to the bipedal robot and Chapter 4 demonstrates the accuracy of kinematics by simulation.
Chapter 5 shows the calculation method of ZMP and then applies it to the lower body of the bipedal robot to verify the dynamic stability while walking. Also the effect of adding torso on dynamic stability is also discussed.
Chapter 6 discusses the effect of variation of torso angle while bipedal robot walking under different circumstances. Here several individual cases have been analyzed thoroughly varying each parameter at a time.
Chapter 7 discusses the conclusions regarding the effectiveness of the torso angle variation in different circumstances while walking in order to maintain dynamic stability and future work.

Chapter 2

Bipedal robot description and trajectory planning

2.1 Bipedal robot description

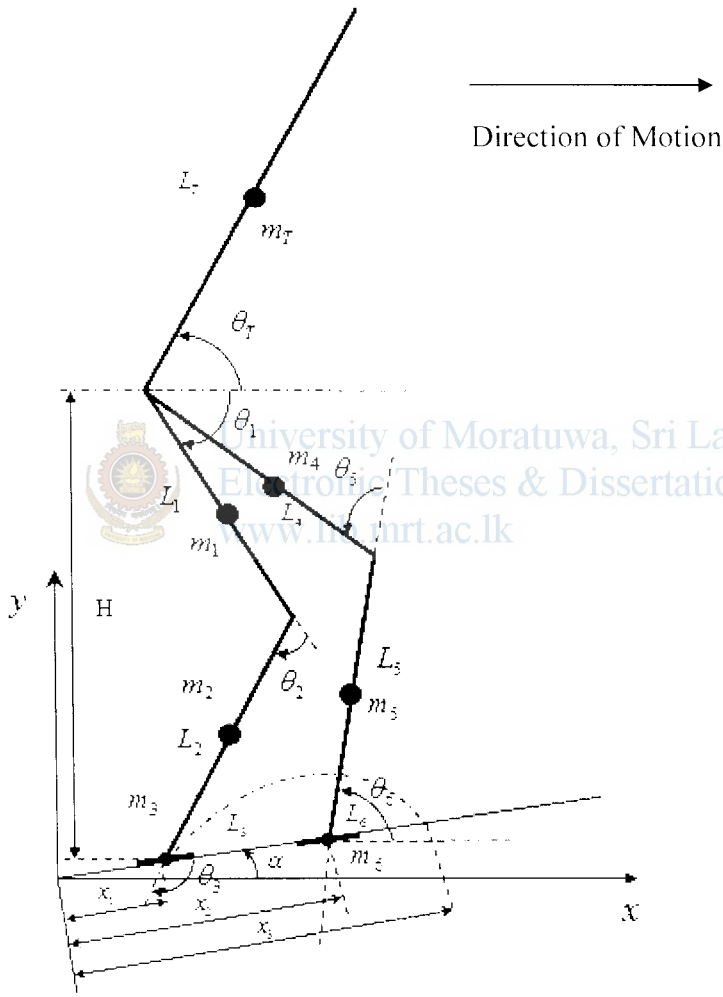


Figure 2.1: 7 links, 7 DoF bipedal robot

The Figure 2.1 shows the schematic of a planer two-legged robot having 7 DoF (two at the ankles, two at the knees and three at the hip), which is moving up along a sloping surface. The robot consists of two ankles, two lower legs, two upper legs and a torso. All the limbs are assumed to have their lumped masses attached at some specific points on them. The joints of the robot are assumed to be consisting of rotary joints. The robot movement has been considered only in sagittal X-Y plane.

2.2 Trajectory Planning

2.2.1 Trajectory

A trajectory is a path with specified quantities of motion. A trajectory is both a spatial and temporal representation of motion. It can be specified either in joint space or in Cartesian space [18].

The various trajectory planning techniques fall into one of the two categories.

1. Joint space techniques
2. Cartesian space techniques

2.2.1.1 Joint space Techniques

This is for the robotic applications involving point to point motion in which motion planning is done at the joint level. The joint- space planning schemes generate time dependent functions (time history) of all joint variables and their first two derivatives to describe the desired motion of the manipulator.

The important features of joint space trajectory planning are,

- Joint angle locations, velocities and accelerations should be continuous functions of time so as to ensure smooth motion.
- The joint polynomial function should not be computationally intensive
- Non-smooth trajectories and other similar undesirable effects are minimized.

2.2.1.2 Cartesian space Techniques

This is for the robotic applications that require continuous path motion. The Cartesian space-planning techniques provide time history of the location, velocity, and acceleration of the end-effector with respect to the base. The corresponding joint variables and their derivatives are computed, using inverse kinematics.

2.2.2 Selection of Trajectories for the Bipedal Robot

2.2.2.1 Trajectory for swing leg

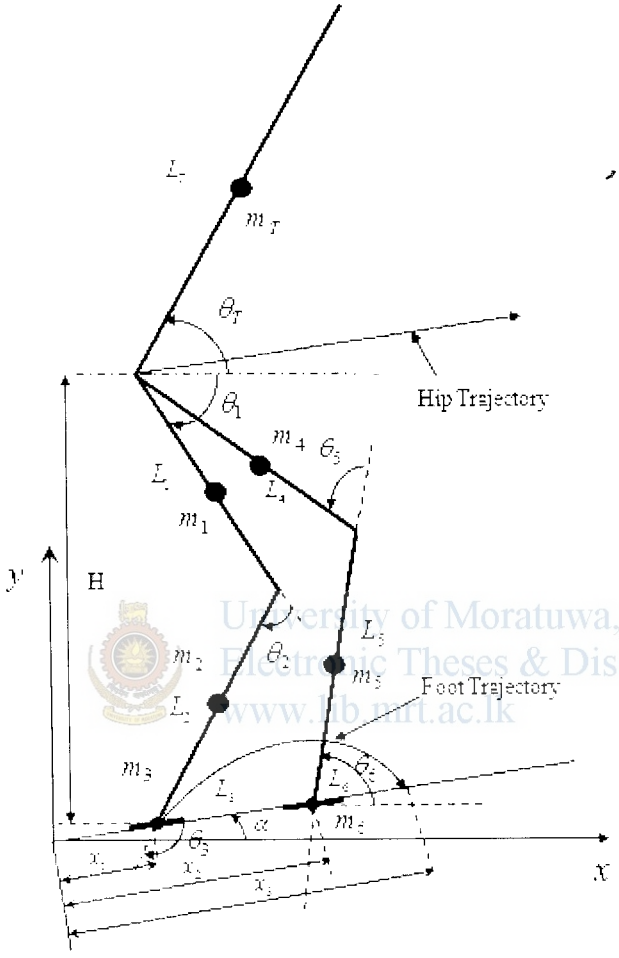


Figure 2.2: Swing leg and Hip trajectories

For the swing leg of bipedal robot the Cartesian space is preferred because the foot trajectory is a predetermined path.

The Figure 2.2 shows bipedal robot with torso. The swing leg joint angles $\theta_1, \theta_2, \theta_3$ and stance leg joint angles θ_5, θ_6 are clearly shown in the diagram. All link lengths and masses are marked against the respective links. The origin of coordinate frame was placed at the beginning of the slope and distances x_1, x_2 indicates the initial positions of swing foot and stance foot while x_3 indicates the final position of the swing foot. The upper body consists of a torso and makes an angle θ_7 with the horizontal axis. The swing foot trajectory and hip trajectory are also indicated.

Here, foot trajectory is assumed to follow cubic polynomial trajectory as shown in Figure 2.2 and defined as,

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \qquad (2.1)$$

Where y represents the height of the swing foot (ankle joint) at a distance x from the starting point and c_0, c_1, c_2, c_3 are the coefficients, whose values are determined using the following boundary conditions.

- at $x = x_1 \cos\alpha$, $y = x_1 \sin\alpha$, (2.2)
- at $x = (x_1 \cos\alpha + x_2 \cos\alpha)/2$, $y = x_2 \sin\alpha + L_3/2$, (2.3)
- at $x = (x_2 \cos\alpha + x_3 \cos\alpha)/2$, $y = x_3 \sin\alpha + L_3/2$, (2.4)
- at $x = x_3 \cos\alpha$, $y = x_3 \sin\alpha$ (2.5)

Applying the boundary conditions (2.2), (2.3), (2.4) and (2.5) in (2.1),

$$x_1 \sin\alpha = c_0 + c_1 x_1 \cos\alpha + c_2 x_1^2 \cos^2\alpha + c_3 x_1^3 \cos^3\alpha \qquad (2.6)$$

$$x_2 \sin\alpha + (L_3/2) = c_0 + c_1 (x_1 \cos\alpha + x_2 \cos\alpha)/2 + c_2 (x_1 \cos\alpha + x_2 \cos\alpha)^2/4 + c_3 (x_1 \cos\alpha + x_2 \cos\alpha)^3/8 \qquad (2.7)$$

$$x_3 \sin\alpha + (L_3/2) = c_0 + c_1 (x_2 \cos\alpha + x_3 \cos\alpha)/2 + c_2 (x_2 \cos\alpha + x_3 \cos\alpha)^2/4 + c_3 (x_2 \cos\alpha + x_3 \cos\alpha)^3/8 \qquad (2.8)$$

$$x_3 \sin\alpha = c_0 + c_1 x_3 \cos\alpha + c_2 x_3^2 \cos^2\alpha + c_3 x_3^3 \cos^3\alpha \qquad (2.9)$$

The equations (2.6), (2.7), (2.8) and (2.9) has been solved writing a MATLAB program A.1 and indicated in Appendix A. Using this program another MATLAB program A.2 indicated in Appendix A has been written to calculate the required foot trajectory coefficients c_0, c_1, c_2 and c_3 numerically by inputting values for x_1, x_2, x_3, α and L_3.

2.2.2.2 Trajectory for hip

The hip trajectory has chosen as a straight line parallel to the slope as shown in Figure 2.2 and defined as ,

$$y = (\tan\alpha)\, x \qquad (2.10)$$

2.2.2.3 Relationship between swing leg movement and hip movement

There should be a relationship between swing leg movement and hip movement for continuous walking. In order to maintain cyclic gait the final position of swing leg should be same as the initial position of the stance leg. After studying the human walking pattern the following formula has been formulated.

Hip movement for Single Step ($x^{"}$) =0.5* Step Length (2.11)

2.2.3 Selection of Joint angle Trajectory for the Bipedal Robot

The selection of a single polynomial for the entire joint path depends on the number of constrains imposed and the type of motion desired. The minimum number of constrains for a smooth motion between two points are

(i) Initial position,
(ii) Initial velocity,
(iii) Final position and
(iv) Final velocity.

With four constrains, a third-degree polynomial with four coefficients can be used. If in addition, with two acceleration constrains fifth-degree polynomial can be used. However, if the degree of polynomial is high (>5), it becomes computationally intensive and it tends to cause extraneous motion.

To ensure smooth functioning, the variation of joint angles of the bipedal robot is assumed to follow fifth-order polynomial as shown below:

$$q_i(t) = a_{i0} + a_{i1}t + a_{i2}t^2 + a_{i3}t^3 + a_{i4}t^4 + a_{i5}t^5 \qquad (2.12)$$

Where $a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}$ and a_{i5} are the coefficients, whose values are determined using different values of q_i at different intervals of time in a cycle and $i = 1,2,....n$ joints.

The assumed velocity variation with time is shown in Figure 2.3.

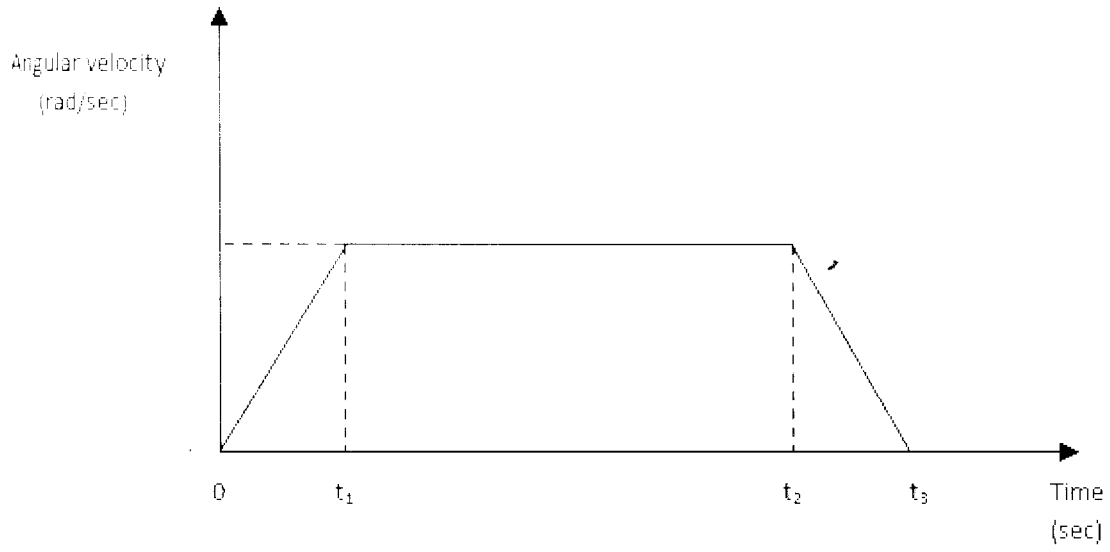


Figure 2.3-Angular velocity variation of the swing leg and stance leg joint angles for half cycle duration

The coefficients $a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}$ and a_{i5} are calculated by referring to the velocity profile shown is Figure 2.3.

Applying following conditions to the equations 2.12

At t=0 initial angular position
At $t=t_3$ final angular position
At t=0 Initial angular velocity
At $t=t_3$ final angular velocity
At $t=t_1$ angular acceleration
At $t=t_2$ angular acceleration

$$q(t_0) = a_{i0} + a_{i1}t_0^2 + a_{i3}t_0^3 + a_{i4}t_0^4 + a_{i5}t_0^5 \qquad (2.13)$$

$$q(t_3) = a_{i0} + a_{i1}t_3^2 + a_{i3}t_3^3 + a_{i4}t_3^4 + a_{i5}t_3^5 \qquad (2.14)$$

$$q(t_0) = 2a_{i1}t_0 + 3a_{i3}t_0^2 + 4a_{i4}t_0^3 + 5a_{i5}t_0^4 \qquad (2.15)$$

$$q(t_3) = 2a_{i1}t_3 + 3a_{i3}t_3^2 + 4a_{i4}t_3^3 + 5a_{i5}t_3^4 \qquad (2.16)$$

$$\ddot{q}(t_1) = 2a_{i2} + 6a_{i3}t_1 + 12a_{i4}t_1^2 + 20a_{i5}t_1^3 \qquad (2.17)$$

$$\ddot{q}(t_2) = 2a_{i2} + 6a_{i3}t_2 + 12a_{i4}t_2^2 + 20a_{i5}t_2^3 \qquad (2.18)$$

Solving the six equations mentioned above the coefficients of the 5th degree polynomial of the respective joint angle can be found.

The MATLAB program A.7 indicated in appendix A has been written to find the coefficients symbolically.
Then using the output of program A.7 another MATLAB program A.8 has been written to calculate coefficients $a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}$ and a_{i5} as indicated in appendix A.

Chapter 3

Kinematics of the bipedal Robot

3.1 Kinematics of the Swing Leg

The kinematics of the lower body has been studied in order to find the relationship between the Cartesian movements and the robot joint angles in order to study robot walking.

3.1.1 Joint angles of Swing Leg

The figure 3.1 shows the swing leg of the bipedal robot.

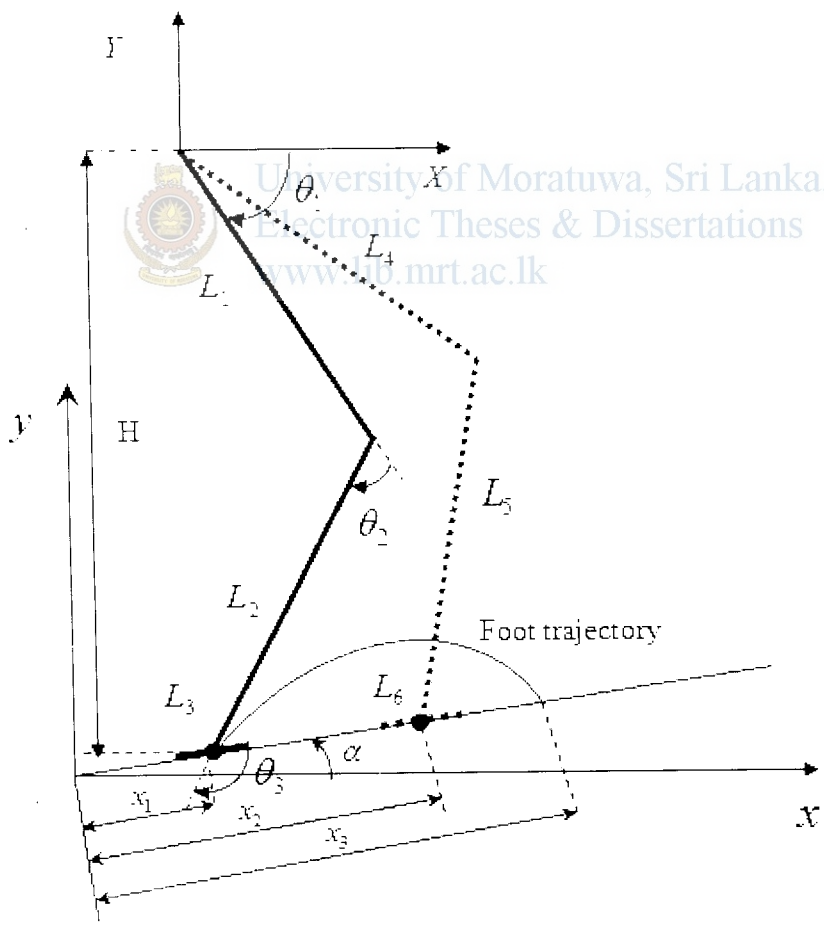


Figure 3.1: Swing leg motion of bipedal robot

The swing leg with link lengths of L_1, L_2 and L_3 and joint angles θ_1, θ_2 and θ_3 moving in highlighted trajectory is illustrated in the Figure 3.1. The stance leg with link lengths of L_4, L_5 and L_6 is indicated in dotted lines. The distance from the initial position of swing foot to the hip is taken as H. The origin of $x-y$ coordinate frame is at the start of the slope. The x_1, x_3 distances indicates swing leg initial and final position from the origin and x_2 is the initial position of the stance leg. The origin of $X-Y$ coordinate system is placed at the hip and used as the reference for the swing leg kinematics.

The Denavit-Hartenberg (D-H) [18] parameters of the swing leg is shown in Table 3.1.The α_{i-1}, a_{i-1}, d_{i-1} and θ_{i-1} have usual Denavit-Hartenberg notation.

Table 3.1: D-H parameters of swing leg

Link number	α_{i-1}	a_{i-1}	d_{i-1}	θ_{i-1}
1	0	0	0	$-\theta_1$
2	0	L_1	0	$-\theta_2$
3	0	L_2	0	$-\theta_3$

The inverse kinematics can be applied to the swing leg considering hip as base. Then the joint angles θ_1, θ_2 and θ_3 can be expressed as [19].

$$\theta_2 = A\tan 2[S_2, C_2] \qquad (3.1)$$

Where,

$$C_2 = [X^2 + Y^2 - L_1^2 - L_2^2 / 2L_1L_2]$$

$$S_2 = \sqrt{1 - C_2^2}$$

$$\theta_1 = A\tan 2\left[\frac{Y}{r}, \frac{X}{r}\right] - A\tan 2(K_2, K_1) \qquad (3.2)$$

Where,

$K_1 = L_1 + L_2C_2$, $K_2 = L_2S_2$ and

$r = \sqrt{K_1^2 + K_2^2}$

By using geometry of the robot,

$$\theta_3 = \theta_1 + \theta_2 + \alpha \qquad (3.3)$$

3.1.2 Coordinate conversion

The coordinates of the of the swing foot has been calculated relative to the Hip coordinate system and the coordinate conversion from the original coordinate system are as follows,

$$X = x - x_1 \cos\alpha \quad (3.4)$$

$$Y = H + x_1 \sin\alpha - y \quad (3.5)$$

3.2 Kinematics of the stance Leg

Kinematics of the stance leg has been studied separately. The Figure 3.2 shows the movement of robot stance leg.

3.2.1 Joint angles of Stance Leg

The Figure 3.2 shows the stance leg of the bipedal robot. The stance leg is highlighted indicating the stance leg joint angles θ_5 and θ_6. The variation of joint angles θ_5 and θ_6 will result the movement of the hip along the trajectory which is parallel to the slope.

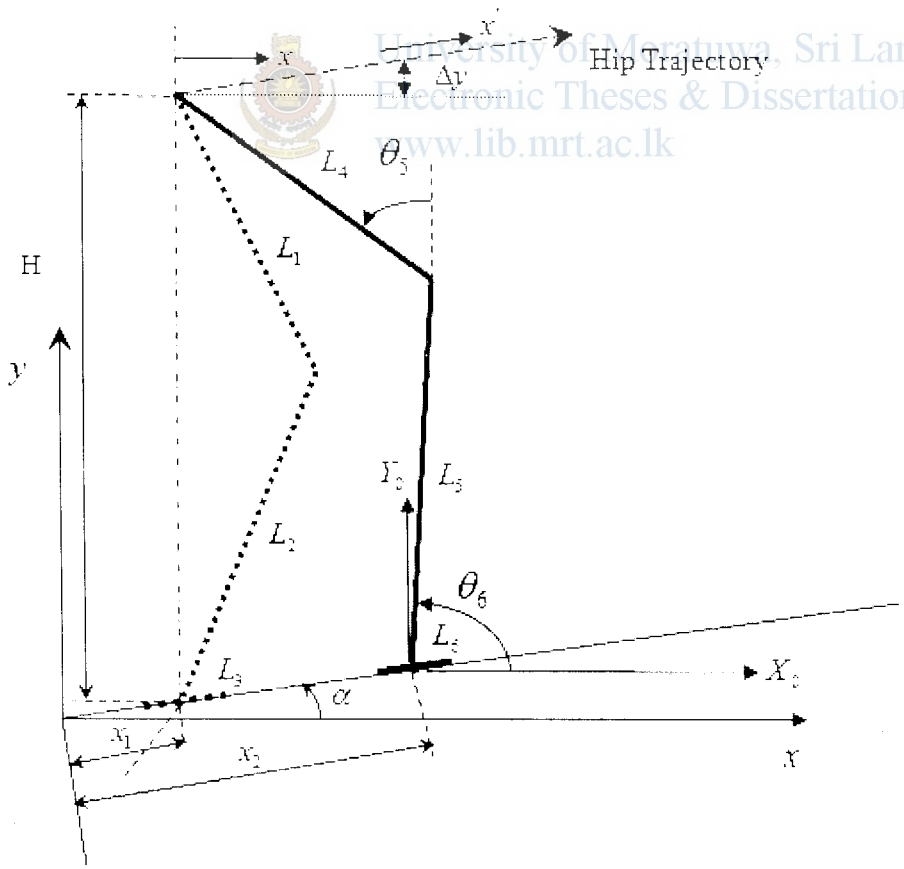


Figure 3.2: Stance leg motion of bipedal robot

The second leg has been considered separately as a three link serial manipulator. A reference coordinate frame located at the foot of the stance foot is considered in deriving kinematics. The D-H parameters of the stance leg is shown in Table 3.2. The swing leg and stance leg are need to be identical in order to walk and the link distances $L_1 = L_4$ and $L_2 = L_5$.

Table 3.2: D-H parameters of stance leg

Link number	α_{i-1}	a_{i-1}	d_{i-1}	θ_{i-1}
1	0	0	0	θ_6
2	0	$L_4(L_1)$	0	θ_5
3	0	$L_5(L_2)$	0	0

Applying inverse kinematics the joint angles θ_5 and θ_6 are found. The respective equation for each joint angle is indicated below.

$$\theta_5 = A\tan 2[S_2, C_2] \qquad (3.6)$$

Where,

$$C_2 = [X_0^2 + Y_0^2 - L_1^2 - L_2^2 / 2L_1L_2]$$

$$S_2 = \sqrt{1 - C_2^2}$$

$$\theta_6 = A\tan 2(Y_0, X_0) - A\tan 2(K_1, K_2) \qquad (3.7)$$

Where,

$$K_1 = L_1 + L_2C_2$$
$$K_2 = L_2S_2$$

3.2.2 Coordinate conversion

The coordinates of the end of link L_4 (Hip) has been calculated relative to the foot of stance leg coordinate system and the coordinate conversion from the original coordinate system are as follows,

$$X_0 = x - (x_2 - x_1)\cos\alpha \qquad (3.8)$$

$$Y_0 = H - (x_2 - x_1)\sin\alpha + \Delta y \qquad (3.9)$$

The second leg has been considered separately as a three link serial manipulator. A reference coordinate frame located at the foot of the stance foot is considered in deriving kinematics. The D-H parameters of the stance leg is shown in Table 3.2. The swing leg and stance leg are need to be identical in order to walk and the link distances $L_1 = L_4$ and $L_2 = L_5$.

Table 3.2: D-H parameters of stance leg

Link number	α_{i-1}	a_{i-1}	d_{i-1}	θ_{i-1}
1	0	0	0	θ_6
2	0	$L_4(L_1)$	0	θ_5
3	0	$L_5(L_2)$	0	0

Applying inverse kinematics the joint angles θ_5 and θ_6 are found. The respective equation for each joint angle is indicated below.

$$\theta_5 = A\tan 2[S_2, C_2] \qquad (3.6)$$

Where,

$$C_2 = [X_0^2 + Y_0^2 - L_1^2 - L_2^2 / 2L_1L_2]$$

$$S_2 = \sqrt{1 - C_2^2}$$

$$\theta_6 = A\tan 2(Y_0, X_0) - A\tan 2(K_1, K_2) \qquad (3.7)$$

Where,

$$K_1 = L_1 + L_2C_2$$
$$K_2 = L_2S_2$$

3.2.2 Coordinate conversion

The coordinates of the end of link L_4 (Hip) has been calculated relative to the foot of stance leg coordinate system and the coordinate conversion from the original coordinate system are as follows,

$$X_0 = x - (x_2 - x_1)\cos\alpha \qquad (3.8)$$

$$Y_0 = H - (x_2 - x_1)\sin\alpha + \Delta y \qquad (3.9)$$

$$\Delta y = (\tan\alpha)\Delta x \qquad (3.10)$$

$$x = x' \cos\alpha \qquad (3.11)$$

3.3 Calculation of joint angles

3.3.1 Calculation of joint angles of swing leg

The following steps are followed to calculate the joint angles θ_1, θ_2.

(i) Coordinates (x, y) of the trajectory path points which the swing leg moves are extracted by finding trajectory coefficients and plotting the trajectory in MATLAB.

(ii) The original coordinates have converted relative to the hip coordinate frame according to the equations (3.4) and (3.5) .

(iii) The equations (3.1) and (3.2) are used to calculate the angles related to each trajectory path point using MATLAB programs A.3 and A.4 indicated in Appendix A.

3.3.2 Calculation of joint angles of stance leg

The following steps are followed to calculate joint angles θ_5, θ_6.

(i) The hip movement distance has been calculated with equation (2.11) and divided equally to the number of swing leg trajectory path points.

(ii) The original coordinates have converted relative to the foot coordinate frame according to the equations (3.8) ,(3.9), (3.10) and (3.11).

(iii) The equations (3.6) and (3.7) are used to calculate the angles related to each trajectory path point using MATLAB programs A.5 and A.6 indicated in Appendix A.

3.3.3 Modification of joint angles of swing leg to compensate hip movement

The joint angles of the swing leg has been calculated in section 3.1.1 assuming the hip is stationary but with the movement of the hip along with stance leg in order to move the swing foot in the same trajectory the hip movement need to be considered in joint angle calculations. Therefore the joint angles calculated considering only the motion of the swing leg in section 3.1.2 need to be recalculated to compensate for the hip movement.

The Figure 3.3 indicates the hip movement along a trajectory parallel to the slope. The horizontal distance the hip moved along x-axis is taken as Δx and the vertical movement along y-axis is taken as Δy. The coordinates (X, Y) used to calculate swing leg joint angles need to recalculate considering the movement of hip making modifications to the equations (3.4) and (3.5).

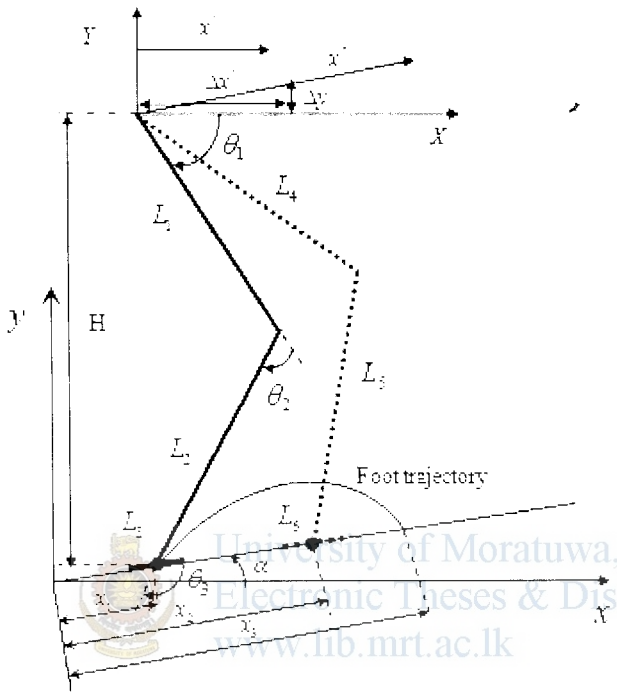


Figure 3.3: The motions of swing leg and stance leg

The modified equations (3.4) and (3.5) to find the (X, Y) coordinates of the trajectory path considering hip movement are as follows.

$$X = (x - x_1 \cos\alpha) - \Delta x \tag{3.12}$$

$$Y = (H + x_1 \sin\alpha - y) + \Delta y \tag{3.13}$$

Chapter 4

Simulation of the bipedal Robot

Simulation of the bipedal robot has been carried out with the results from the previous chapter. The bipedal robot walking on 5 degree slope has been simulated after creating joint angle data. MATLAB software has been used extensively for the joint angle calculations. Each trajectory has been plotted and extracted data points such that the different between two consecutive data points to be minimum in order to get very accurate result. Each simulation involves more than hundred data points for each joint angle variation.

4.1 Simulation Software

The simulation software used in this simulation is "Robo Works "introduced by company called Newtonium. This software is user friendly and has capability to simulate using data files [20].

4.2 Simulation of Swing Leg

The MATLAB programs A.1 and A.2 in Appendix A has been used for the calculation of the coefficients of the equation 2.1 using the parameters mentioned in table 4.1. The body parameters of the bipedal robot are taken approximately equal to the human skeleton.

Table 4.1-Parameters for the swing leg simulation

L_1, L_4	460mm		x_1	100mm
L_2, L_5	480mm		x_2	450mm
L_3, L_6	180mm		x_3	800mm
α	5^0		H	700mm

The foot trajectory after the calculation was.

$$y = 91.6482 + 1.1225x - 0.0012x^2 \qquad (4.1)$$

The Figure 4.1 shows the plot of the calculated trajectory along with the slope.

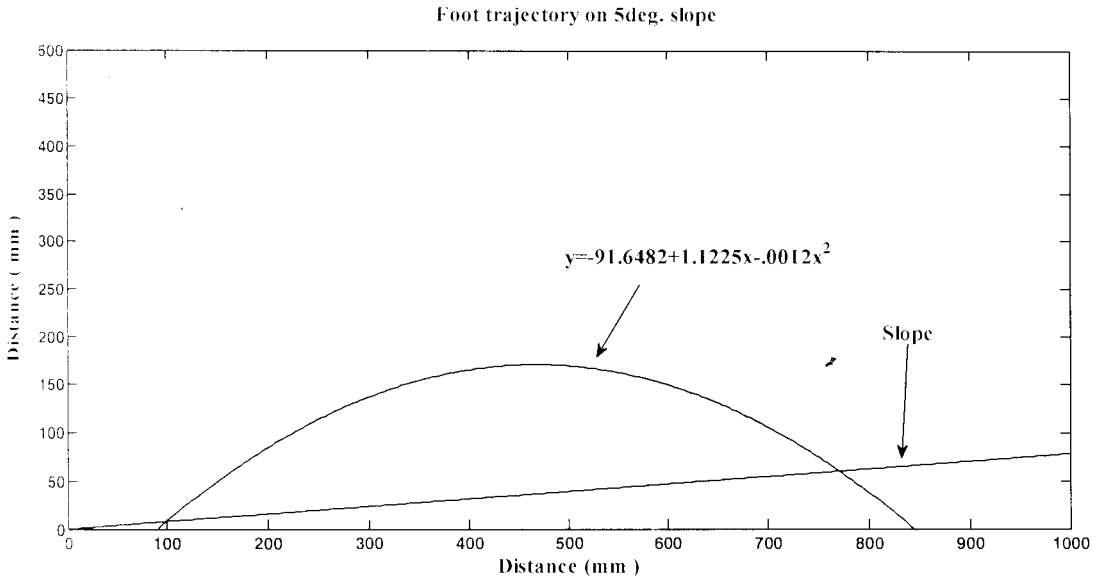


Figure 4.1: Foot trajectory at 5 degree slope

The simulation program E.1 written for the simulation of swing leg is indicated in Appendix E. The data file has been created in .dat format using the angles θ_1, θ_2 and θ_3 calculated and used in the simulation.

The figure 4.2, 4.3, 4.4, 4.5 and 4.6 shows the simulation steps of swing leg simulation.

The figure 4.2 shows the initial position of the swing leg and stance leg. The current data point shown is zero. The monitor window shows the initial values of the swing leg joint angles θ_1, θ_2 and θ_3 .

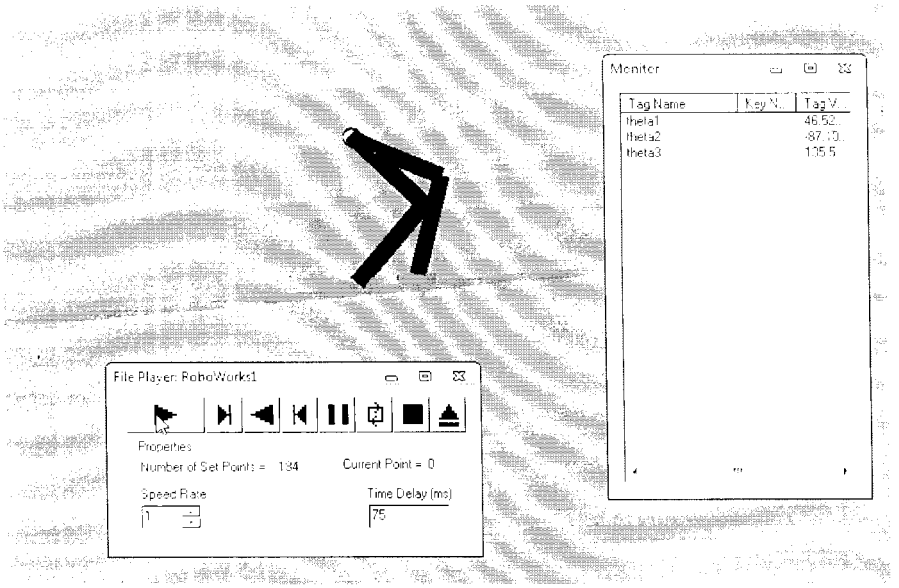


Figure 4.2: Initial pose of the swing leg simulation

The Figure 4.3 shows the swing leg at 42nd data point and the swing leg joint angles are visible in monitor window.

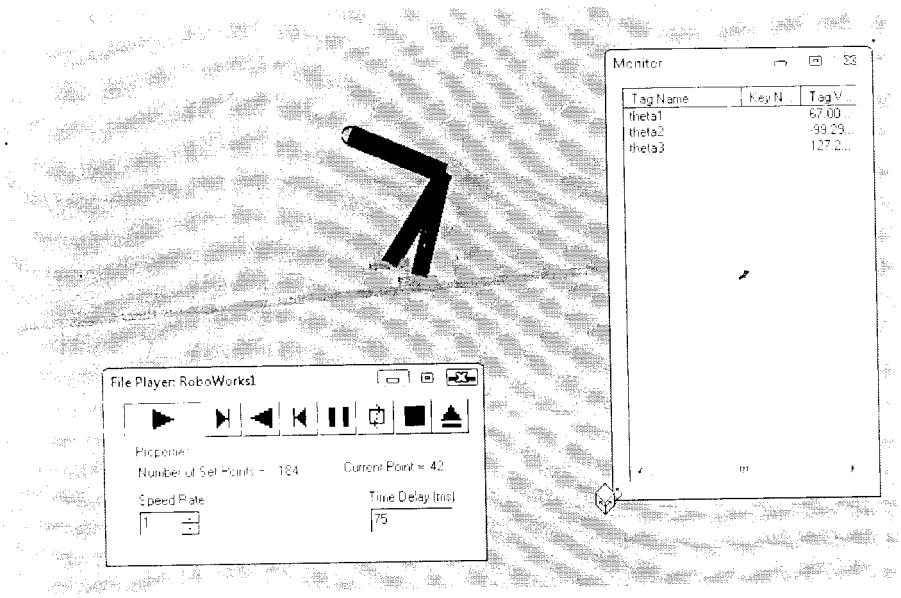


Figure 4.3: Swing leg simulation at 42nd data point

The Figure 4.4 shows another pose of the swing leg simulation at 163th data point.

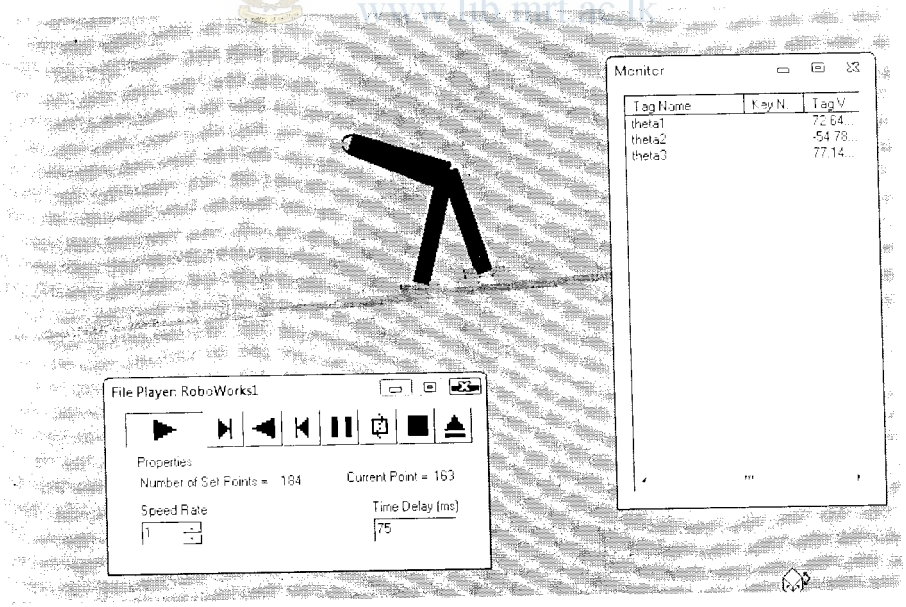


Figure 4.4: Swing leg simulation at 163rd data point

The Figure 4.5 shows the swing leg at the final data point.

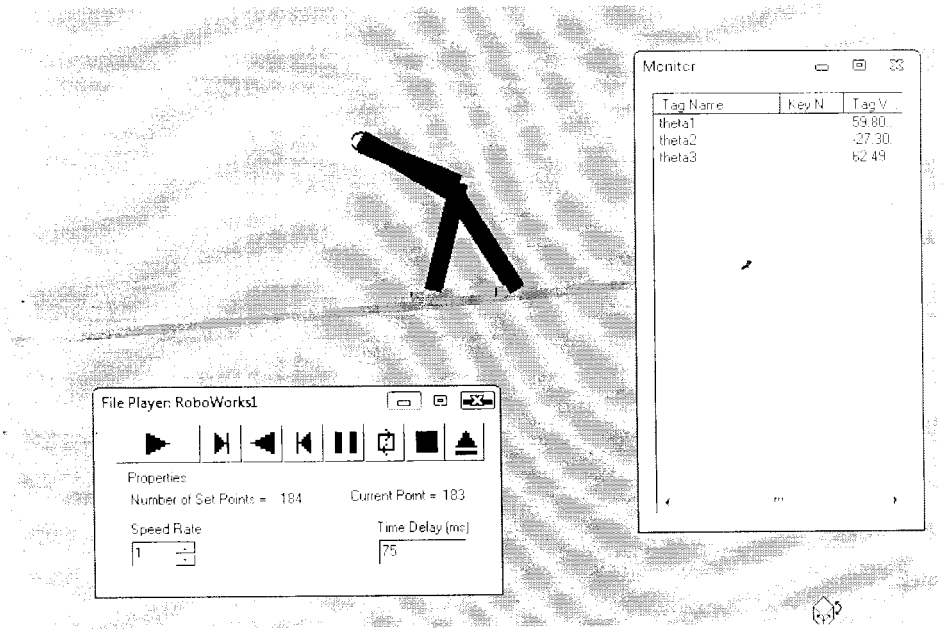


Figure 4.5: Final pose of the swing leg simulation

The Figure 4.6 shows the accuracy of the swing leg simulation by showing all the movements in a single frame. It can be seen that the foot of the swing leg is following the trajectory shown in Figure 4.1. This proves that the accuracy of the joint angle equations.

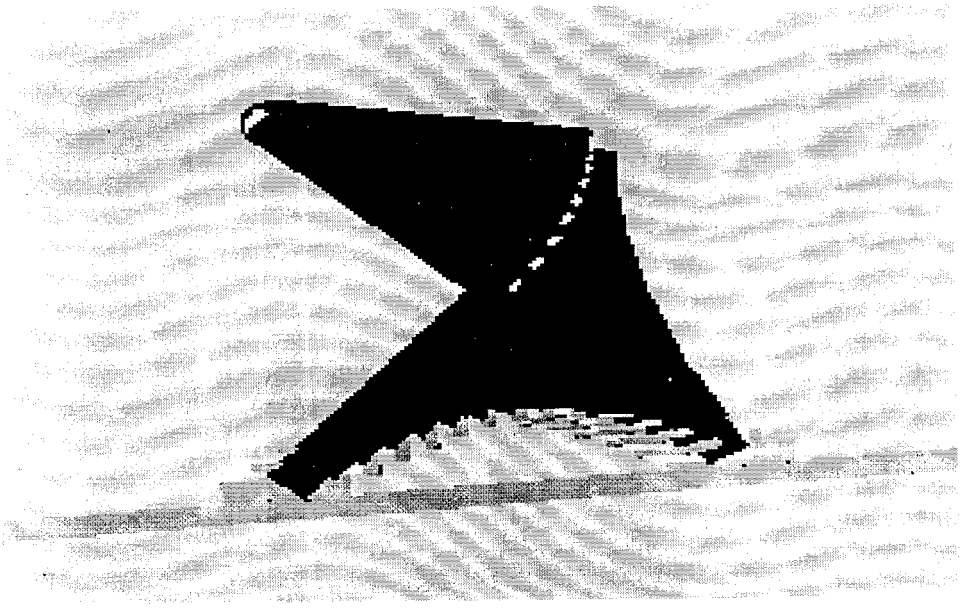


Figure 4.6: The swing leg following predefined trajectory

4.3 Simulation of Stance Leg

The stance leg simulation has been carried out using the calculated joint angles θ_5 and θ_6. A data file in .dat format has been created using joint angle data in order to simulate the stance leg in RoboWorks software.

The Figure 4.7 shows the initial and final positions of the stance leg. The Figure 4.8 shows the complete motion and shows correctness of the joint angle calculation because the hip moves in the hip trajectory planned.

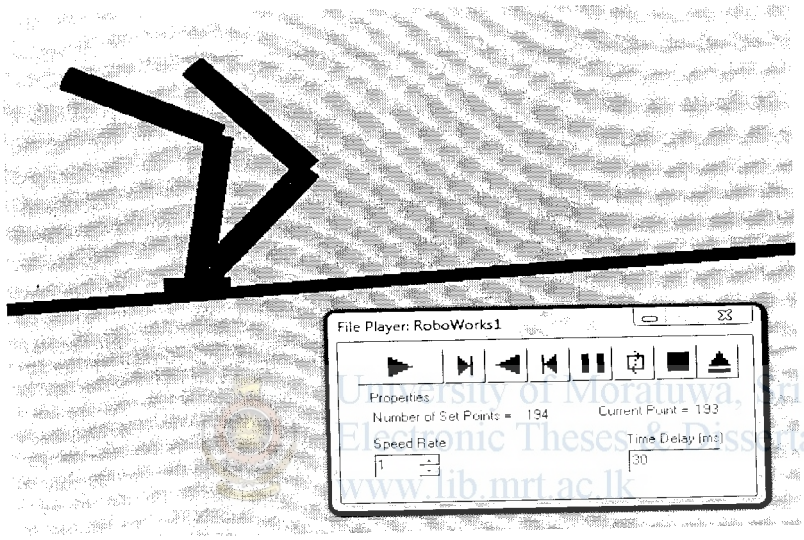


Figure 4.7: The initial and final positions of the stance leg

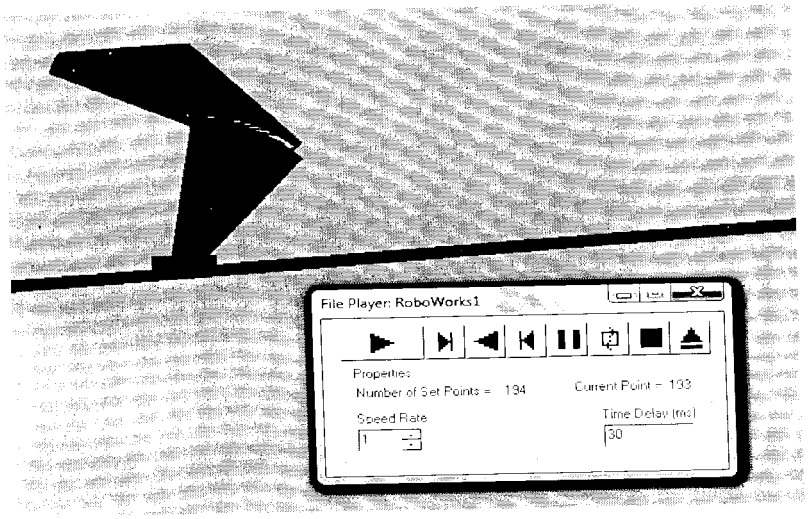


Figure 4.8: The stance leg following the predefined trajectory

4.4 Simulation of both legs

The simulation of both legs has been carried out after modification of angles θ_1 and θ_2 of swing leg as indicated in section 3.3.3 and joint angle data of both legs were used to create data file in .dat format The Figures from 4.9 to 4.13 shows the different poses of the simulation from start to end.

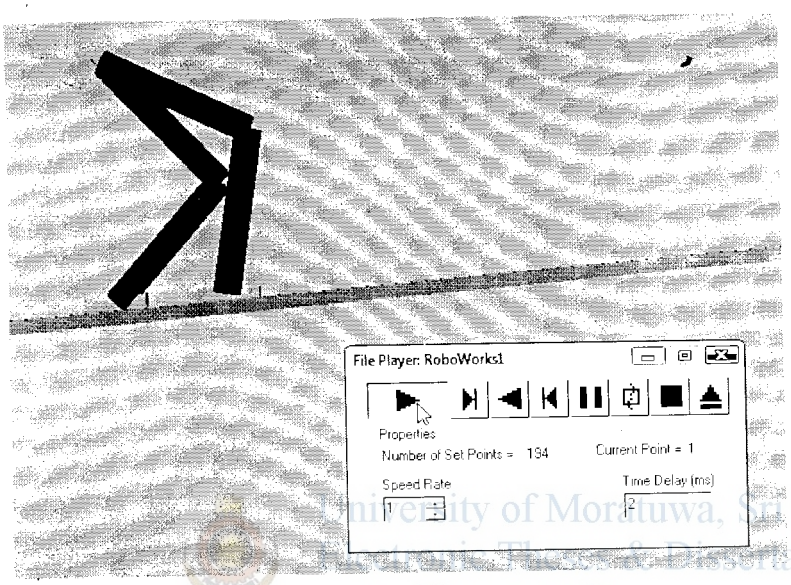


Figure 4.9: Initial position of lower body

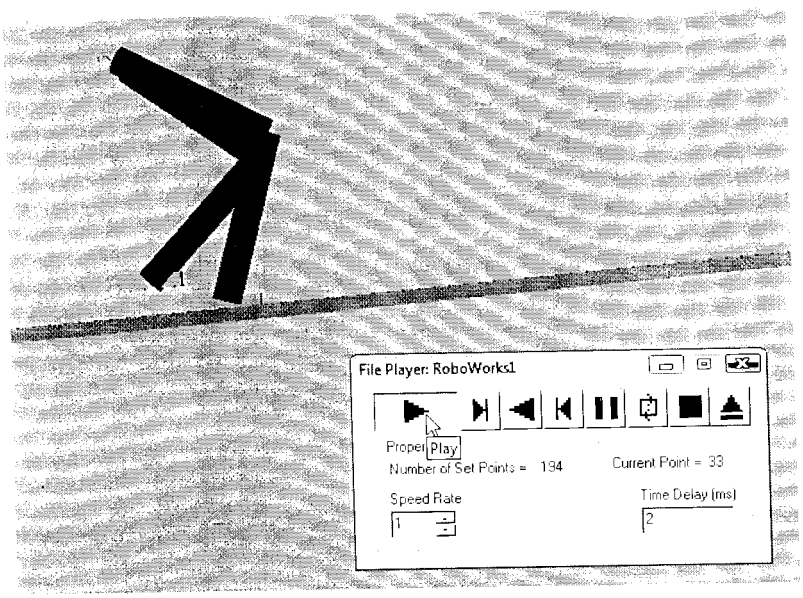


Figure 4.10: Lower body simulation at 33rd data point

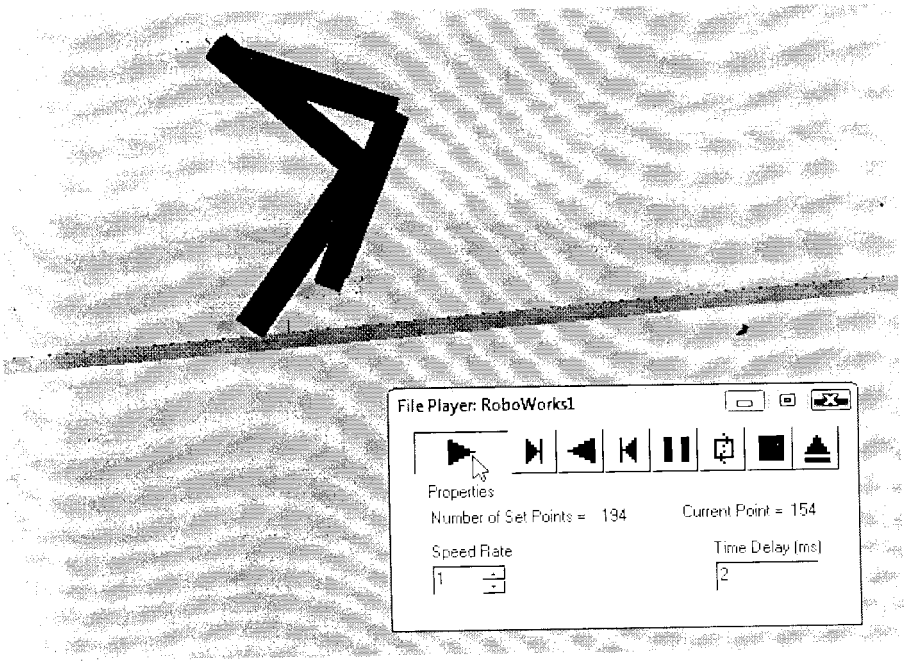


Figure 4.11: Lower body simulation at 154th data point

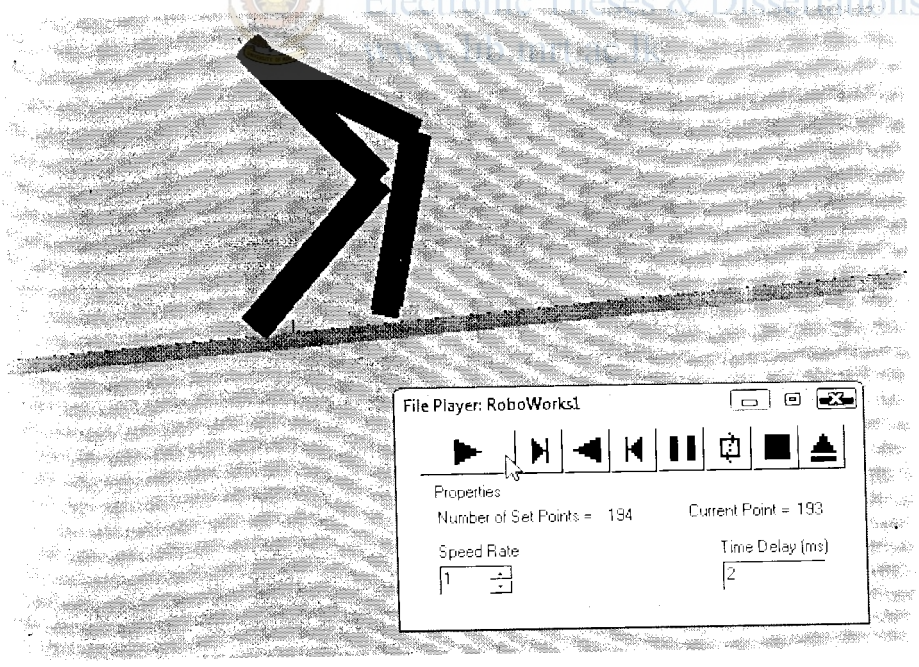


Figure 4.12: Final position of the lower body

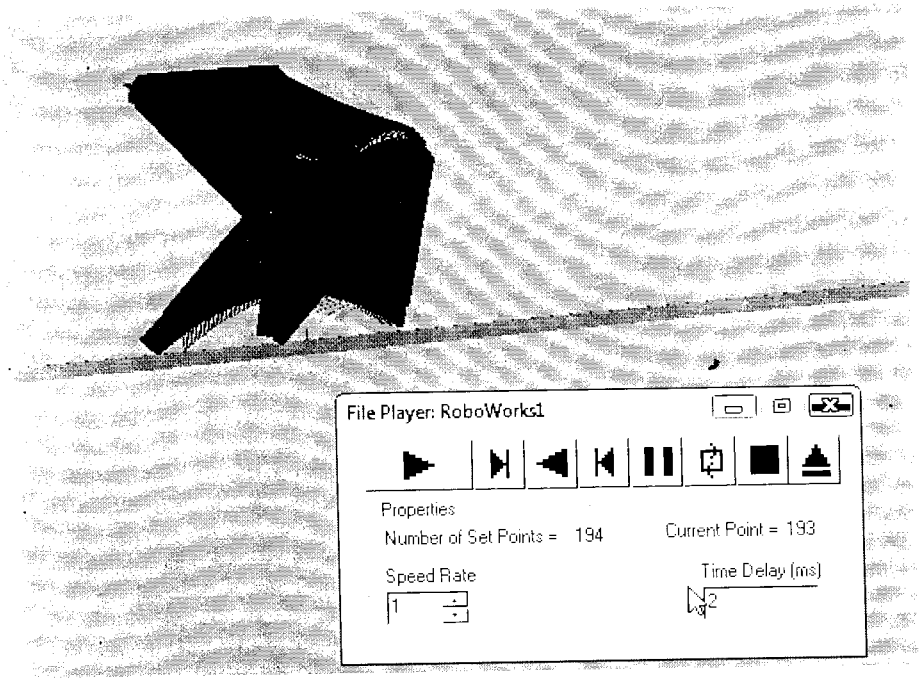


Figure 4.13: Lower body following foot and hip trajectory

The simulation of lower body has been carried out with 193 data points with each data point consisting values of angles $\theta_1, \theta_2, \theta_3, \theta_5$ and θ_6 which are calculated using the joint angle equations derived using inverse kinematics. The Figure 4.9 shows the initial pose of both legs. The Figure 4.10 and Figure 4.11 shows the simulation poses at 33rd and 154th data points. The Figure 4.12 indicates the final pose of the simulation.

In order to maintain cyclic gait the final position of stance leg should be equal to the initial position of the swing leg and this is clearly visible when comparing the Figure 4.9 and Figure 4.12.

The Figure 4.13 shows the swing foot trajectory and hip trajectory both following the predefined trajectories as planned in the section 2.2.2.

4.5 Simulation of Lower body and Torso

The simulation has been carried out after adding torso. The hip consists of 3 rotary joints with two for legs and one for the hip. The angle the torso makes with the horizontal axis is considered as torso angle.

These simulation windows show how the bipedal will negotiate ascending slope of 5 degrees. The Figures 4.14 to 4.17 shows the different simulation phases which shows the beginning, intermediate and finishing stages.

Figure 4.14: Initial position of the bipedal robot

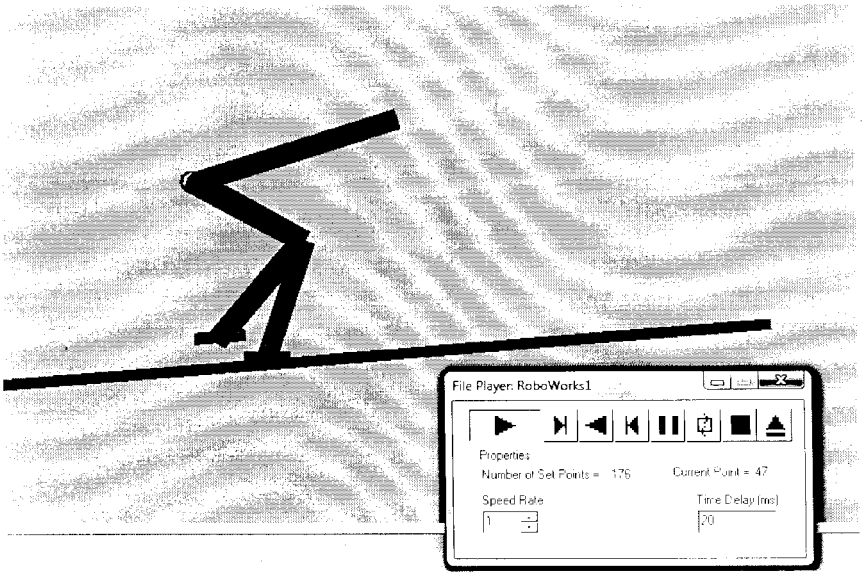


Figure 4.15: Simulation at 47th data point

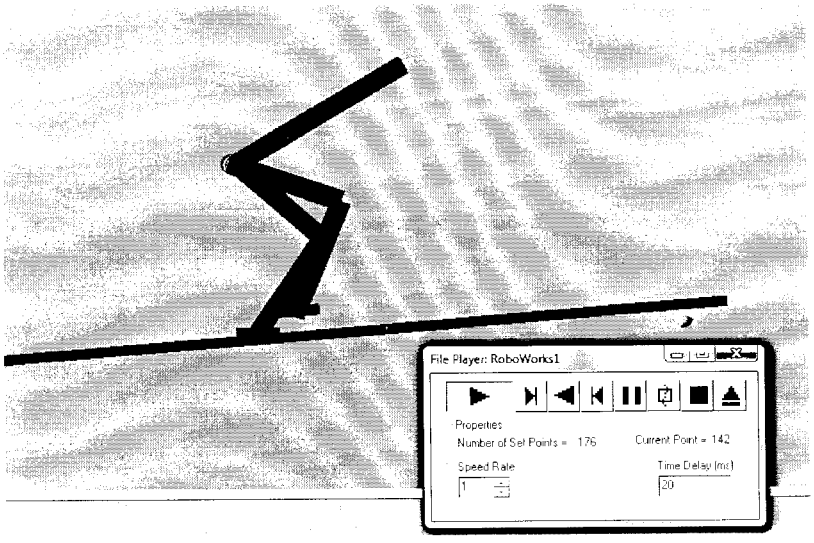


Figure 4.16: Simulation at 142nd data point

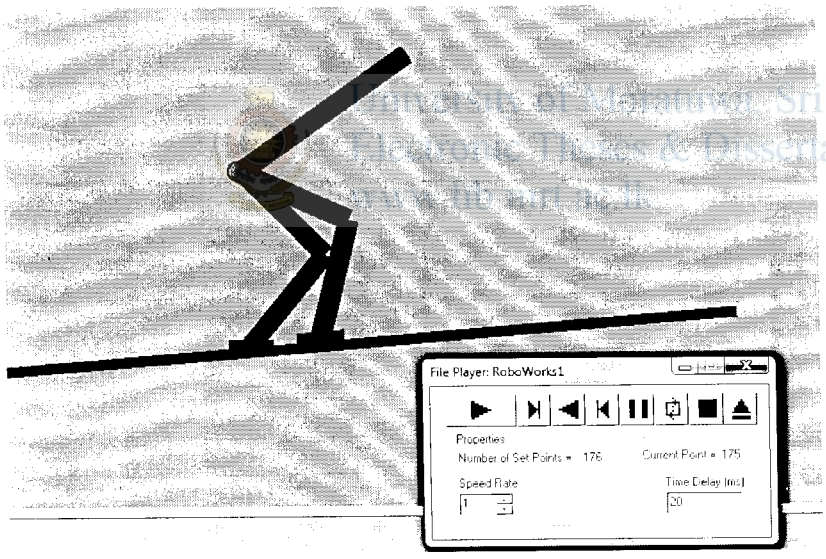


Figure 4.17: The final position of motion

The figure 4.18 shows all the motions of swing leg, stance leg and torso in a single window to illustrate trajectories of motion. This simulation result verifies the kinematics of the bipedal robot and validates this application in real environment. This simulation has been carryout with large data points and hence the swing foot and hip moves in the predefined trajectory smoothly.

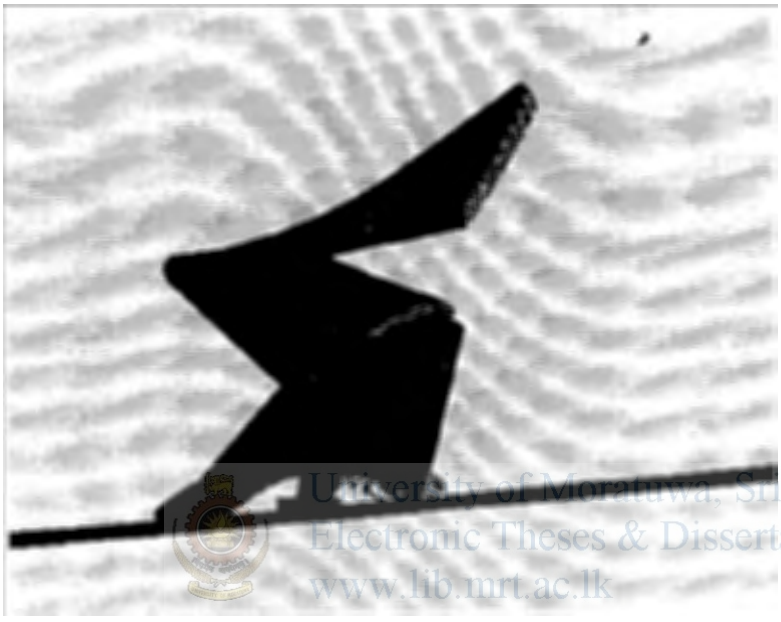


Figure 4.18: The swing leg, hip and torso movement

Chapter 5

Calculation of ZMP and Dynamic Stability

5.1 General ZMP Equation for the bipedal robot

The distance of the ZMP from the ankle joint of the supporting leg of the bipedal robot in the direction of motion can be determined from the following equation [21].

$$X_{zmp} = \Sigma_{i=1}^{4}(I_i \omega_i + m_i X_i (Y_i - g) - m_i X_i Y_i) / \Sigma_{i=1}^{4} m_i (Y_i - g) \qquad (5.1)$$

I_i -Moment of Inertia of i^{th} link (kg/m^2)

ω_i -Angular acceleration of link I (rad/s^2)

m_i -Mass of i^{th} link (Kg)

x_i, y_i -Coordinate of the i^{th} lumped mass

g -Acceleration due to gravity (m/s^2)

Y_i -Acceleration of link i in y-direction (m/s^2)

X_i -Acceleration of link i in x-direction (m/s^2)

5.2 Dynamic Stability Margin (DSM) of bipedal robot

The figure 5.1 shows the foot of the stance foot and marked with distance X_{ZMP}. The walking becomes dynamic stable if the ZMP falls between the dynamic stability margin of stance foot.

The dynamic stability region (DSM) = $(x_2 - L_6/2)$ to $(x_2 + L_6/2)$ (5.2)

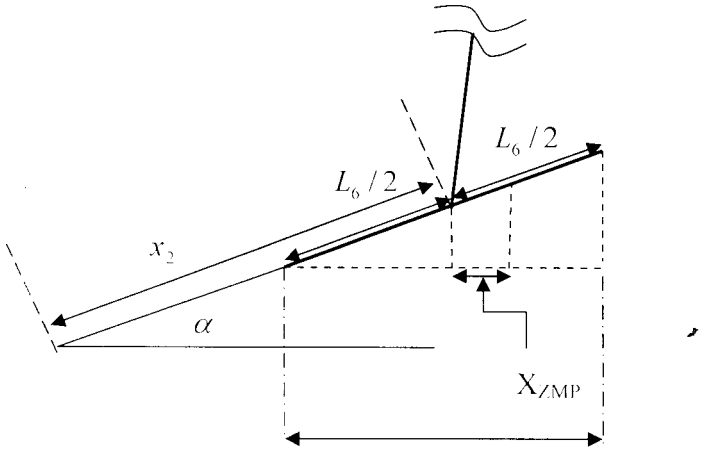


Figure 5.1-Dynamic stability safe margin

5.3 Application of ZMP Equation for the bipedal robot

The equation 5.1 can be applied to find the ZMP of bipedal robot in swinging phase at each trajectory point in order to find dynamic stability while walking.

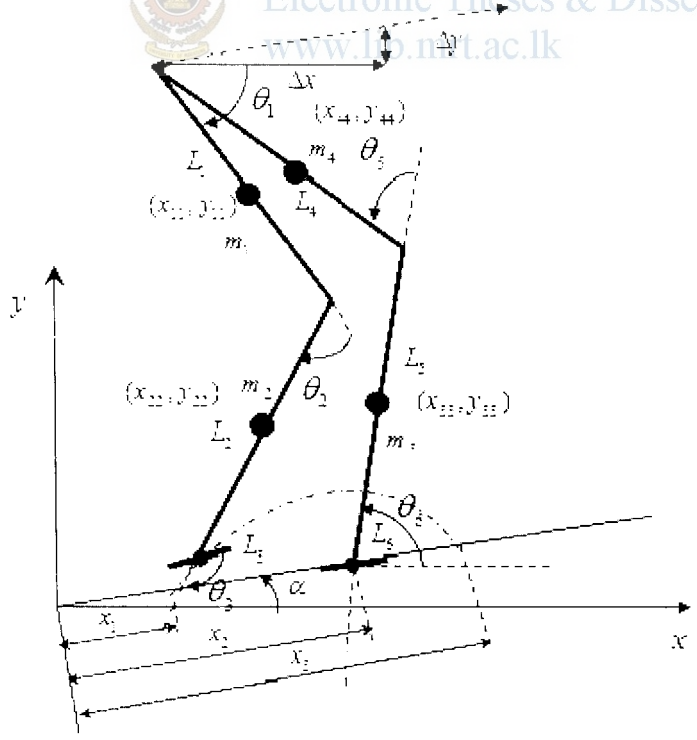


Figure 5.2-Lower body of bipedal robot

The Figure 5.2 shows the masses, lengths of each link and coordinates of the centre of mass of each link.

Here to have identical legs.

$m_1 = m_4$, $m_2 = m_3$, $L_1 = L_4$, $L_2 = L_5$.

5.3.1 Mass centre coordinates of links

The following equations were formulated considering the motion of the bipedal robot.

$$x_{c1} = x_1 \cos\alpha + \Delta x + 0.5L_1 \cos\theta_1 \quad (5.3)$$

$$y_{c1} = y + 0.5L_1 \sin\theta_1 + L_2 \sin(\pi - (\theta_1 + \theta_2)) \quad (5.4)$$

$$x_{c2} = x + 0.5L_3 \cos(\pi - (\theta_1 + \theta_2)) \quad (5.5)$$

$$y_{c2} = y + 0.5L_2 \sin(\pi - (\theta_1 + \theta_2)) \quad (5.6)$$

$$x_{c3} = x_3 \cos\alpha + 0.5L_2 \cos\theta_6 \quad (5.7)$$

$$y_{c3} = x_3 \sin\alpha + 0.5L_2 \sin\theta_6 \quad (5.8)$$

$$x_{c4} = \Delta x + 0.5L_1 \cos(\pi - (\theta_5 + \theta_6)) + x_1 \cos\alpha \quad (5.9)$$

$$y_{c4} = x_3 \sin\alpha + L_2 \sin\theta_6 + 0.5L_1 \sin(\pi - (\theta_5 + \theta_6)) \quad (5.10)$$

5.3.2 Angular accelerations of links

The coefficients of the fifth-order joint angle polynomial found by writing a MATLAB programs indicated in Appendix A.7 and A.8 indicated in Appendix A to solve six equations indicated in section 2.2.3. This program has been used to find each joint angle polynomial separately.

The angular acceleration of joint i is given by:

$$\alpha_i(t_2) = 2a_{i1} + 6a_{i3}t_2 + 12a_{i4}t_2^2 + 20a_{i5}t_2^3 \quad (5.11)$$

5.3.3 Accelerations of links

The recursive Newton-Euler formulation has been carried out to find the link accelerations of each link of bipedal robot and mentioned below. [22]

$$V_{1x} = V_x C_1 + (V_y - g) S_1 - 0.5 L_1 \dot{\theta}_1^2 \quad (5.12)$$

$$V_{1y} = V_x S_1 + (V_y - g) C_1 - 0.5 L_1 \ddot{\theta}_1 \quad (5.13)$$

$$V_{1x} C_2 + V_{1y} S_{21} - 0.5 L_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (5.14)$$

$$-V_{1x} S_2 + V_{1y} C_2 + 0.5 L_2 (\ddot{\theta}_1 + \ddot{\theta}_2) \quad (5.15)$$

$$-gS_3 - 0.5 L_2 \dot{\theta}_3^2 \quad (5.16)$$

$$-gC_3 - 0.5 L_2 \ddot{\theta}_3 \quad (5.17)$$

$$V_{4x} = L_2 \ddot{\theta}_3 S_4 - L_2 \dot{\theta}_3^2 C_4 - L_1 \dot{\theta}_3^2 - L_1 \dot{\theta}_4^2 - 2 L_1 \dot{\theta}_3 \dot{\theta}_4 - g S_{34} \quad (5.18)$$

$$V_{4y} = L_2 \ddot{\theta}_3 C_4 - L_1 \ddot{\theta}_3 + L_1 \ddot{\theta}_4 + L_2 \dot{\theta}_3^2 S_4 - g C_{34} \quad (5.19)$$

$$V_{1x} + 0.5 L_1 (\ddot{\theta}_3 + \ddot{\theta}_4) \quad (5.20)$$

$$V_{4y} - 0.5 L_1 (\ddot{\theta}_3 + \ddot{\theta}_4) \quad (5.21)$$

5.3.4 Inertia of links

The robot is considered to be made up of slender bars.

The moment of inertia of i th link $I_i = 1/12 m_i L_i^2$ (5.22)

here m_i -mass of link i

L_i -length of link i

5.3.5 Building MATLAB program to calculate ZMP

A MATLAB program A.9 has been formulated to calculate ZMP in X direction combining the equations indicated in sections 5.3.1, 5.3.2, 5.3.3 and 5.3.4 These MATLAB programs for each case is indicated in appendix A.
This MATLAB function can be used to calculate ZMP in X direction at any instant.

5.4 Calculation method of ZMP

The steps of calculation of ZMP are indicated below. For all the ZMP calculations the following method has been followed.
The foot has not been considered for the ZMP calculation due to the smaller mass compared to other links hence its effect on the dynamic stability of robot is assumed to be negligible.

(i) Deciding values for $L_1, L_2, L_3, L_4, L_5, L_6, \alpha, H, x_1, x_2, x_3$.

(ii) Calculation of trajectory coefficients using the Matlab program A.2.

(iii) Plotting of trajectory and extracting trajectory coordinates x, y.

(iii) Calculation of coordinates X,Y using equations 3.4 and 3.5.

(iv) Calculation of hip trajectory coordinates $\Delta x, \Delta y$ using equations 3.10 and 3.11.

(v) Modification of X,Y using equations 3.12 and 3.13.

(vi) Calculation of joint angle θ_2 using MATLAB program A.3.

(vii) Calculation of joint angle θ_1 using MATLAB program A.4.

(viii) Calculation of joint angle θ_6 using MATLAB program A.6.

(ix) Calculation of joint angle θ_5 using MATLAB program A.5.

(x) Deciding values for $m_1, m_2, m_3, m_4, T, t_0, t_1, t_2, t_3, x''$.

(xi) Calculation of coefficients of each joint angle trajectory $a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}$ and a_{i5} using MATLAB program A.8.

(xii) Calculation of ZMP using relevant MATLAB program indicated in Appendix A using the data calculated in previous steps.

5.5 ZMP of the Lower body of bipedal robot

ZMP values for single step have been calculated using the MATLAB program A.9 using trajectory data using parameters in Table 5.1.

Step Length (SL) $= x_3 - x_1$

Step Time (T) $= t_3 - t_0$

Table 5.1-Parameters for the calculation of ZMP of lower body

l_1, l_4	460 mm	x_1	100 mm
l_2, l_3	480 mm	x_2	450 mm
l_5, l_6	180 mm	x_3	800 mm
α	5^0	H	700 mm
g	9.81m/s^2	m_1, m_4	5.0 Kg
$x^{\cdot}$	350 mm	m_2, m_3	5.2 Kg
T	5 sec	t_0	0 sec
t_1	1.25 sec	t_2	3.75 sec
t_3	5.00 sec	SL	700 mm

5.6 ZMP variation of the Lower body of bipedal robot

The ZMP values obtained have been potted and shown in Figure 5.3. The dynamic stability margin has been calculated according to the equation 5.2.

DSM =360mm to 540mm

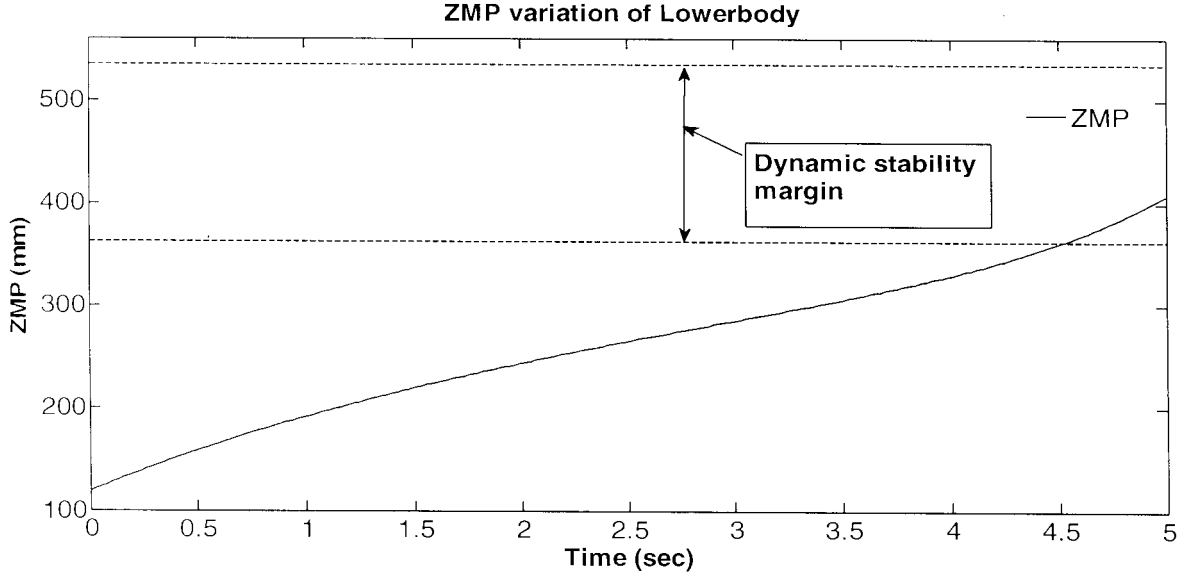


Figure 5.3-ZMP variation of Lower body of the bipedal robot

According to the Figure 5.3 lower body motion is highly unstable during most of the step time therefore it is evident that lower body alone cannot achieve dynamic stability.

5.7 Adding Torso

As highlighted in section 5.6 it is impossible to achieve dynamic stability alone with lower body. Therefore in hoping to gain stability a torso has been added as an upper body. The figure 5.4 indicates the bipedal robot with torso.

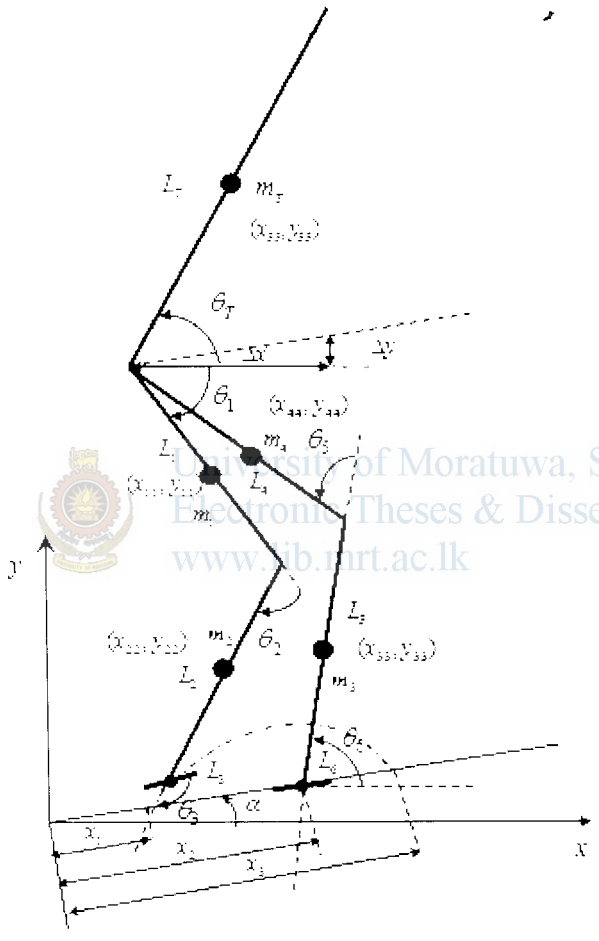


Figure 5.4-Bipedal robot with Torso

5.7.1 Mass centre coordinates of Torso

The equation derived considering the geometry of the bipedal robot to calculate the mass centre coordinates of the torso and as follows,

$$x_{ss} = x_1 \cos\alpha + \Delta x + 0.5 L_5 \cos(\theta_T) \qquad (5.23)$$

$$= H + \Delta y + 0.5 L_5 \sin(\theta_7) \qquad (5.24)$$

5.7.2 Angular acceleration of Torso

Angular acceleration of torso is assumed to be very small compared to the accelerations of other links hence it is considered as zero for the ZMP computations.

5.7.3 Acceleration of Torso

The recursive Newton-Euler formulation has been carried out to find the Torso acceleration and mentioned below. [22]

$$= V_{4x} \cos\theta_7 - V_{4y} \sin\theta_7 - 0.5 L_7 (\theta_5 + \theta_6 + \theta_7) \qquad (5.25)$$

$$= V_{4x} \sin\theta_5 + V_{4y} \cos\theta_7 + 0.5 L_7 (\theta_5 + \theta_6 + \theta_7) \qquad (5.26)$$

5.8 Defining torso angle

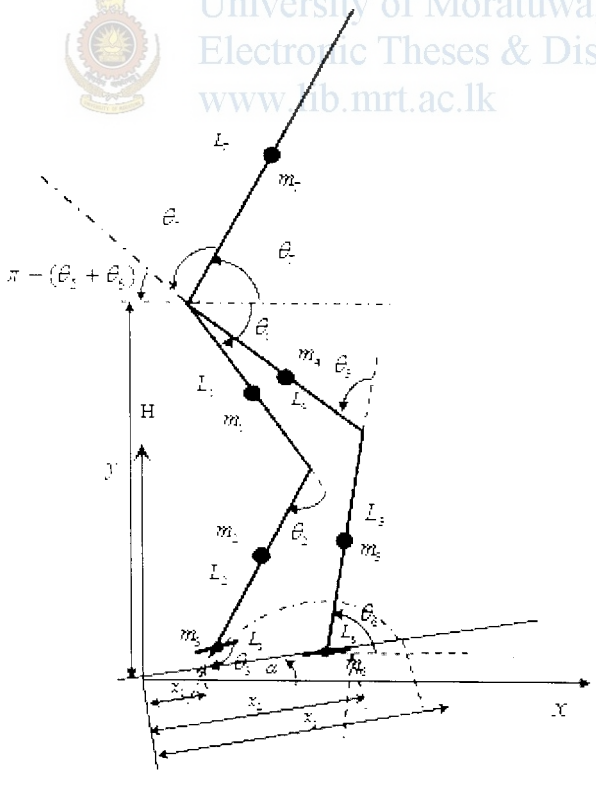


Figure 5.5-Illustration of torso angle

As shown in Figure 5.5 the angle θ_T has been introduced as Torso angle.

Torso angle $(\theta_T) = \theta_5 + \theta_6 - \theta_7$ (5.27)

In order to keep θ_T at fixed value while walking it is necessary to adjust the angle θ_7 accordingly to suppress the variation of θ_1.

Therefore the angle θ_7 can be calculated using equation 5.27 at each data point to calculate the respective ZMP value at that torso angle.

$\therefore \quad \theta_7 = \theta_5 + \theta_6 - \theta_T$

5.9 ZMP of the bipedal robot with torso

ZMP values for single step have been calculated using the matlab program A.10 in Appendix A using trajectory data using data in Table 5.2

Here torso angle with respective to the horizontal axis, θ_T is taken as 10^0.

Table 5.2- Parameters for the calculation of ZMP of bipedal robot with torso

L_1, L_4	460 mm		x_1	100 mm
L_2, L_5	480 mm		x_2	450 mm
L_3, L_6	180 mm		x_3	800 mm
α	5^0		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
$x^{\cdot}$	350 mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		θ_T	10^0
SL	700 mm		L_T	800 mm
m	35Kg			

5.10 ZMP variation of the bipedal robot with torso

The ZMP values obtained have been plotted and shown below. The dynamic stability margin has been calculated according to the equation 5.2.

DSM =359mm to 538mm

From the Figure 5.6 it is evident that by adding a torso the bipedal robot has gained stability for almost 70 % of the step time.

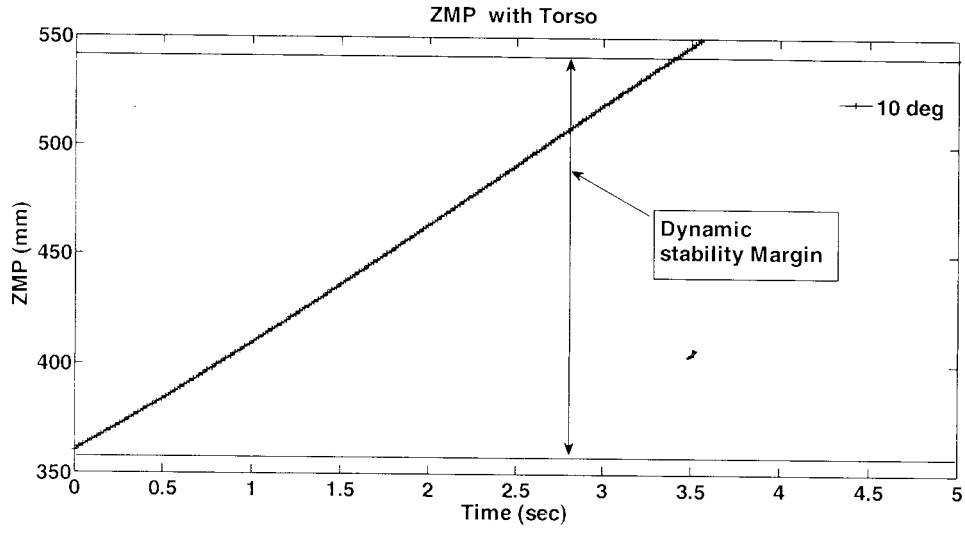


Figure 5.6-ZMP of bipedal robot with Torso

5.11 ZMP of the bipedal robot with different torso angles

ZMP values for single step have been calculated using the MATLAB program A.10 using trajectory data and using data in Table 5.3.

Here six different torso angles (θ_T) are considered.

Table 5.3- Parameters for the calculation of ZMP of bipedal robot with different torso angles

l_1, l_4	460 mm		x_1	100 mm
l_2, l_5	480 mm		x_2	450 mm
l_3, l_6	180 mm		x_3	800 mm
α	5^0		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
$x^{"}$	350 mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		θ_T	10^0,15^0,20^0,25^0,30^0,35^0
SL	700 mm		L_T	800 mm
m_T	35Kg			

5.12 ZMP variation of the bipedal robot with different torso angles

The ZMP values obtained have plotted as shown in Figure 5.7. The dynamic stability margin has been calculated according to the equation 5.2.

DSM =359mm to 538mm

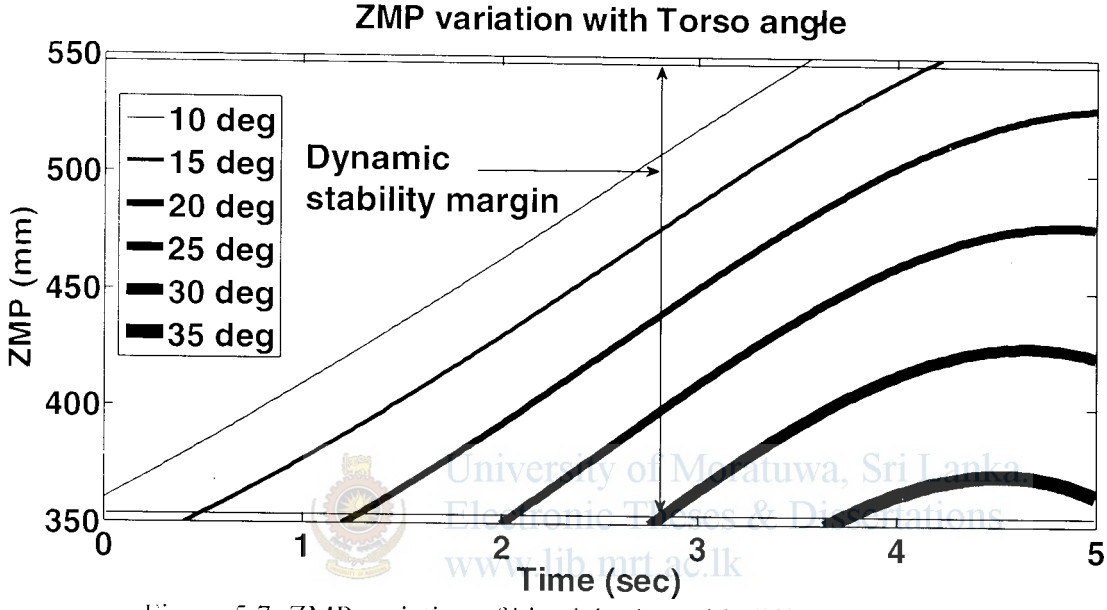


Figure 5.7: ZMP variation of bipedal robot with different torso angles

The figure 5.7 shows six lines representing ZMP for different torso angles. It is clear observing graph that the bipedal robot can keep dynamic stability by varying torso angle from 10^0-20^0.

Chapter 6

Behavioral analysis of bipedal robot

The previous chapter shows the capability of gaining dynamic stability of bipedal robot by adding a torso as upper body and varying torso angle.
In this chapter the effectiveness of gaining dynamic stability by varying torso is further analyzed by studying different cases varying single parameter in each case such as slope angle, step length, step time, etc.
Also the joint angle variation is analyzed and presented because smooth joint angle variation is important to minimize jerks while walking.
All the calculations in this chapter have been done according to the method indicated in section 5.4.
The full step length is considered as 700 mm which is approximately equal to the human step length when walking.

6.1 Effect of Torso angle on dynamic stability on various slopes

Here bipedal robot walking on three slopes having slope angles 5^0, 10^0 and 15^0 are investigated. The data relevant to the ZMP calculation of all slope angles are indicated in Table 6.1.

Table 6.1- Parameters for the calculation of ZMP of bipedal robot at different slopes

l_1, l_4	460 mm		x_1	100 mm
l_2, l_5	480 mm		x_2	450 mm
l_3, l_6	180 mm		x_3	800 mm
SL	700 mm		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
$x^{"}$	350 mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		L_T	800mm
m_T	35Kg			

6.1.1 Bipedal robot walking on 5^0 slope

The bipedal robot walking on 5^0 slope is considered. The ZMP calculation has been carried out for different torso angles using the MATLAB program A.10 in appendix A and the ZMP values are plotted and shown in Figure 6.1.

DSM 359mm to 538mm

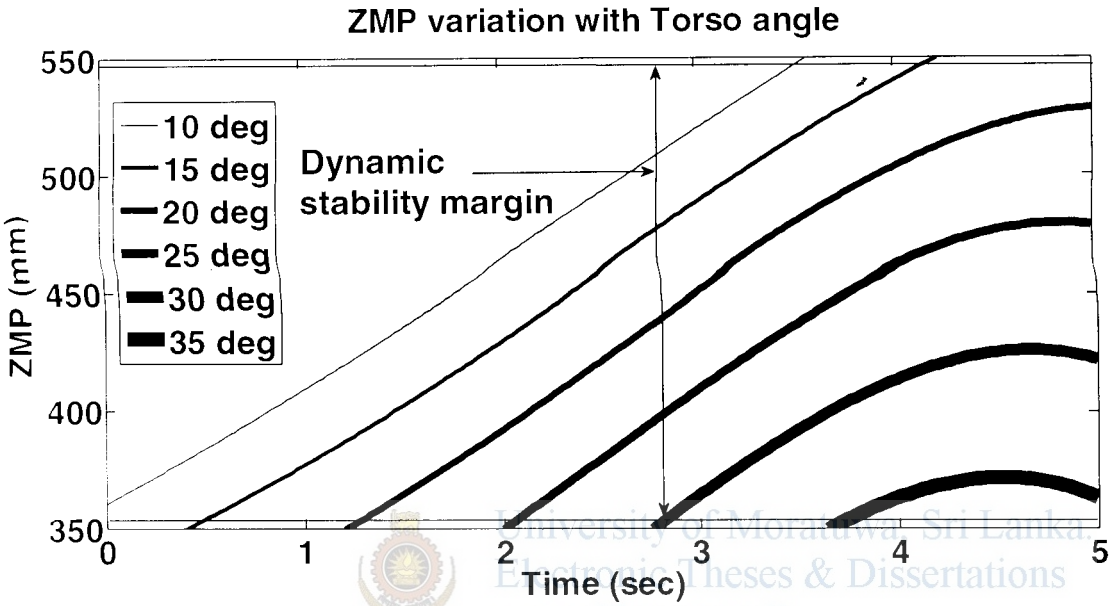


Figure 6.1-ZMP variation with torso angle at 5 deg. slope with full step length

According to the Fig.6.1 torso angle only need to be vary from 10-35 degrees in order to maintain the dynamic stability during its swinging time. There is a greater change in ZMP value for torso angle. It is also observed when increasing Torso by 5^0 step ZMP improved significantly. Therefore it is evident that by adding a torso it is possible to maintain the dynamic stability by adjusting torso angle accordingly. It is observed for a certain angle the ZMP stays in dynamic stability region more than half of the step time therefore to maintain dynamic stability torso does not need to vary quickly.

In all three cases of slope angle 5^0,10^0 and 15^0 the robot can maintain dynamic stability by varying the torso angle therefore this robot can negotiate steep slopes without falling.

6.1.2 Bipedal robot walking on 10^0 slope

The bipedal robot walking on 10^0 slope is considered. The ZMP calculation has been carried out for different torso angles using the MATLAB program A.11 in appendix A and the ZMP values are plotted and shown in Figure 6.2. According to the Figure 6.2 the torso angle can be varied from 7-35 degrees in 10 degrees ascending slope.

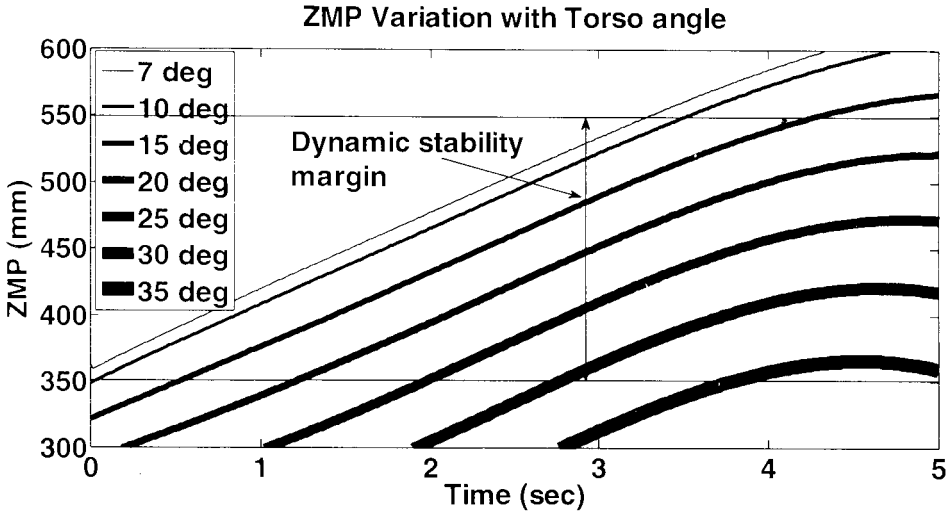


Figure 6.2-ZMP variation with torso angle at 10 deg. slope with full step length

6.1.3 Bipedal robot walking on 15^0 slope

The ZMP calculation has been carried out for different torso angles using the MATLAB program A.12 in appendix A and the ZMP values are plotted and shown in Figure 6.3. According to the Figure 6.3 the torso angle can be varied from 10-35 degrees in 15 degrees ascending slope.

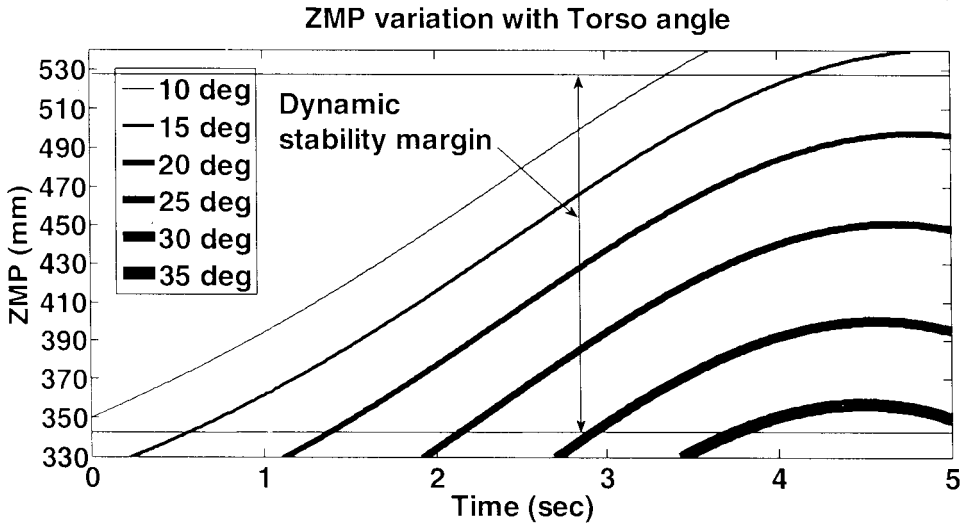


Figure 6.3-ZMP variation with torso angle at 15 deg. slope with full step length

6.2 Effect of Torso angle on dynamic stability at different steps

The bipedal robot walking on 5^0 ascending slope while varying step length is considered for the following cases. The data relevant to the ZMP calculation of different steps are indicated in Table 6.2.

Table 6.2- Parameters for the calculation of ZMP of bipedal robot at different steps

L_1, L_4	460 mm		x_1	100 mm
L_2, L_5	480 mm		m_T	35Kg
L_3, L_6	180 mm		L_T	800mm
SL	700mm,350mm,175mm		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
x	350mm,175mm,87.5mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec			

6.2.1 Walking on full step

The ZMP calculation has been carried out for different torso angles using the MATLAB program A.10 in appendix A and the ZMP values are plotted and indicated in Figure 6.4.

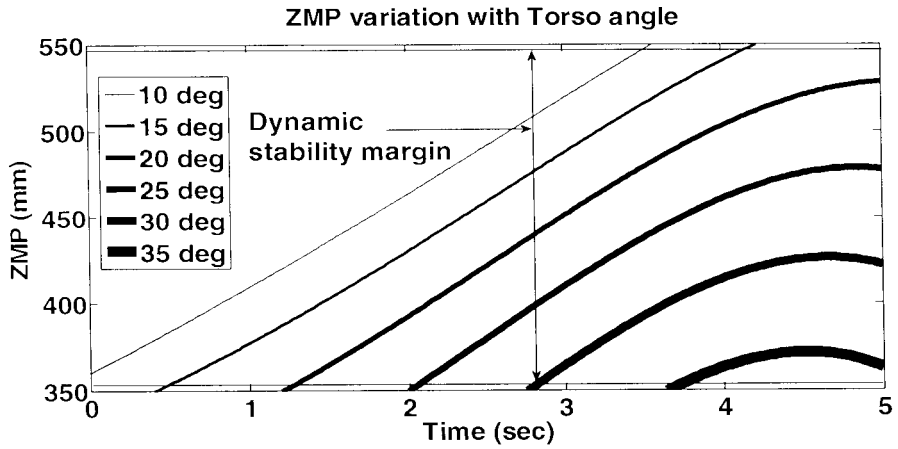


Figure 6.4-ZMP variation with torso angle at 5 deg. slope with full step length

6.2.2 Walking on half step

The ZMP calculation has been carried out for different torso angles using the MATLAB program A.13 in appendix A and the ZMP values are plotted and indicated in Figure 6.5.

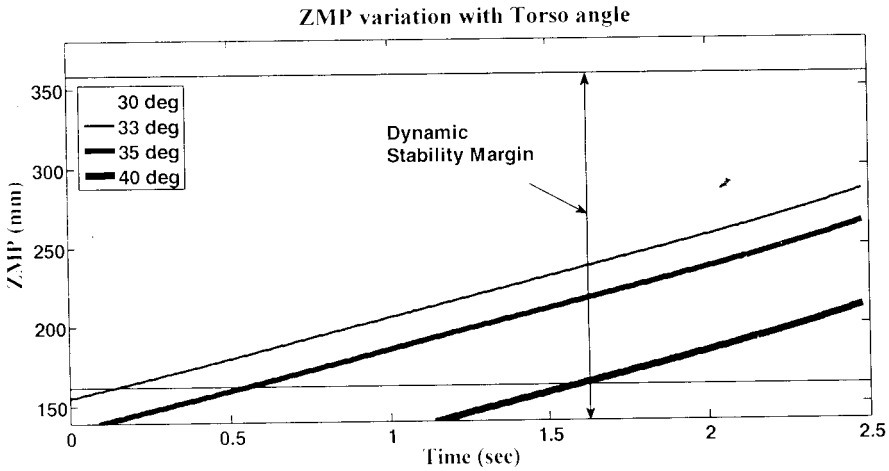


Figure 6.5-ZMP variation with torso angle at 5 deg. slope with half step length

6.2.3 Walking on quarter step

The ZMP calculation has been carried out for different torso angles using the MATLAB program A.14 in appendix A and the calculated ZMP values are plotted and indicated in Figure 6.6.

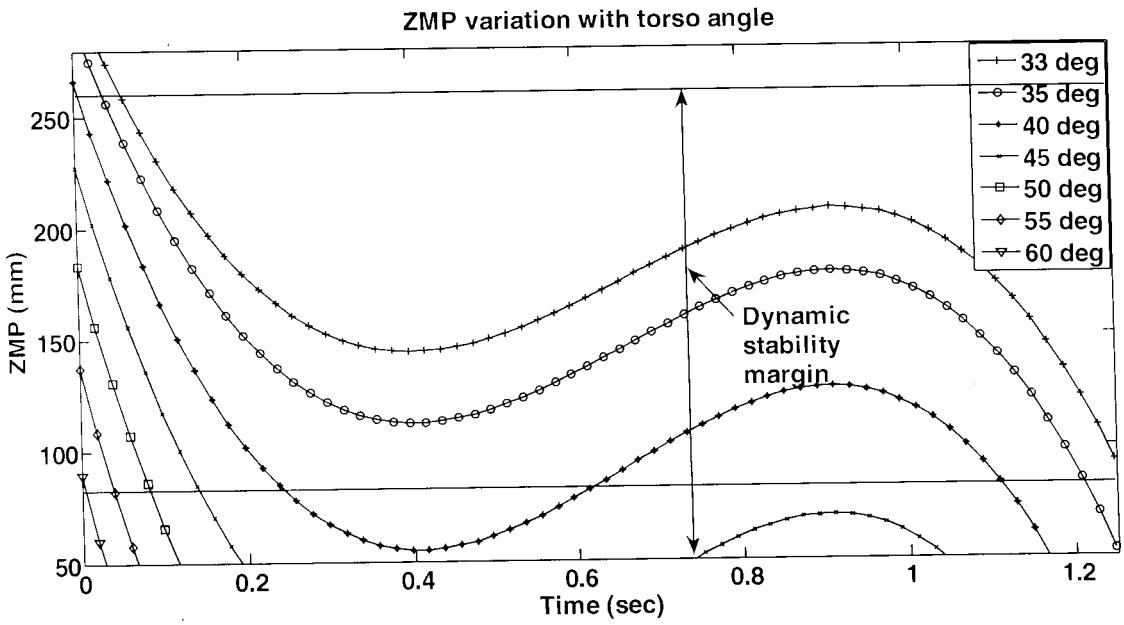


Figure 6.6-ZMP variation with torso angle at 5 deg. slope with quarter step length

above three cases the step length has been varied to find out the behavior of dynamic stability of the bipedal robot. After observing all three graphs it is observed that the torso angle can be increased in steep slopes while keeping dynamic stability. In Figure 6.6 it can be seen that the torso angle can be increased up to 60 degrees while maintaining the dynamic stability. The reason for this is when the bipedal robot is taking smaller steps it is easy to maintain the dynamic stability because ZMP is closer to the dynamic stability margin rather than taking large steps.

6.3 Effect on dynamic stability at different step durations

The ZMP calculation has been carried out keeping torso angle at 10 degrees varying step time. The MATLAB program A.10 in appendix A has been used with data in Table 6.3 and the ZMP values are plotted and shown in Figure 6.7.

Table 6.3- Parameters for the calculation of ZMP of bipedal robot at different step times

l_1, l_4	460 mm	x_1	100 mm
l_2, l_5	480 mm	x_2	450 mm
l_3, l_6	180 mm	x_3	800 mm
SL	700 mm	H	700 mm
g	9.81m/s^2	m_1, m_4	5.0 Kg
x'	350 mm	m_2, m_3	5.2 Kg
l_7	800mm	m_7	35Kg

In Figure 6.7 ZMP variation of this bipedal robot has been tested by varying the time taken to swing the leg. It can be seen the robot is very unstable if it walks very slowly. But as long as it walks fast the ZMP improves. Therefore with this result it is possible to state that the bipedal robot is capable of fast walking keeping its dynamic balance

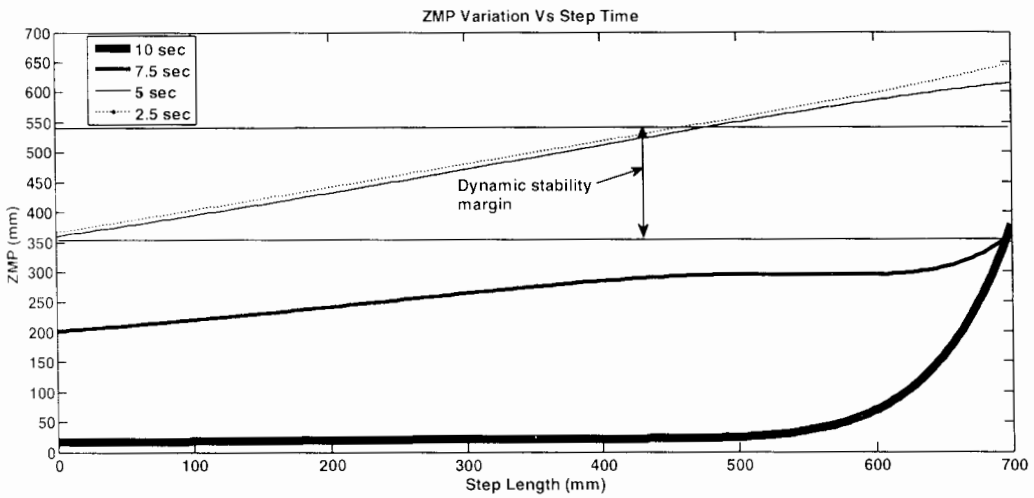


Figure 6.7-ZMP variation with torso angle at 5 deg. slope with full step length varying step time

6.4 Effect of Torso mass on dynamic stability of robot

In this section the effect of the dynamic stability of the bipedal robot on varying torso mass is analyzed. Two different cases at two different torso angles have been analyzed.

Table 6.4- Parameters for the calculation of ZMP of bipedal robot with different masses of Torso

l_1, l_4	460 mm		x_1	100 mm
l_2, l_5	480 mm		x_2	450 mm
l_3, l_6	180 mm		x_3	800 mm
SL	700 mm		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
x	350 mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		L_T	800mm

6.4.1 Dynamic stability with different masses on torso at torso angle of 10^0

The bipedal robot walking on 5^0 ascending slope with full step has been considered for this case. The MATLAB program A.10 in appendix A has been used with data in Table 6.4 and the ZMP values are plotted and shown in Figure 6.8.

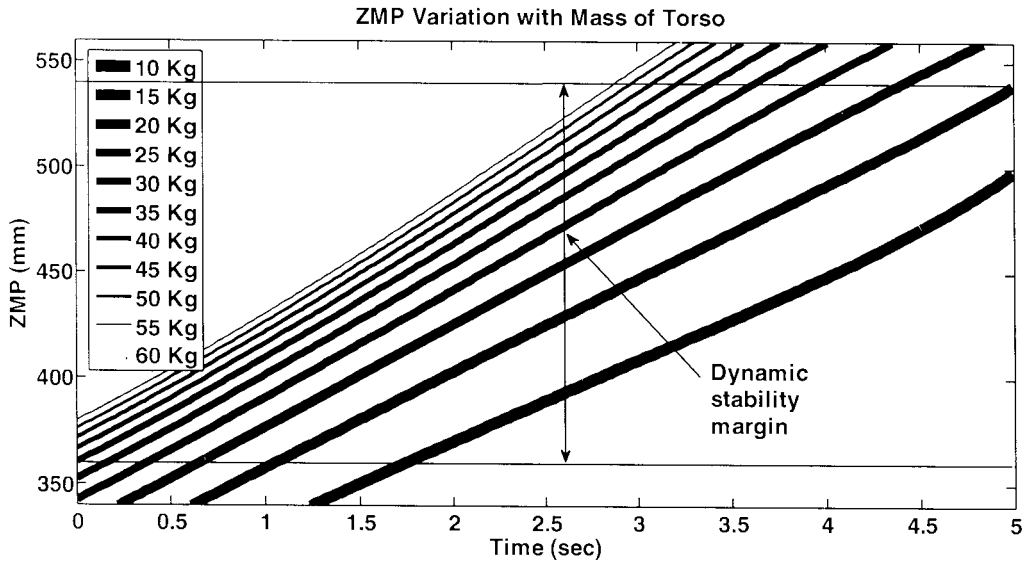


Figure 6.8-ZMP variation keeping torso angle at 10 deg. and varying torso mass

6.4.2 Dynamic stability with different masses on torso at torso angle of 15^0

The bipedal robot walking on 5^0 ascending slope with full step has been considered for this case. The MATLAB program A.10 in appendix A has been used with data in Table 6.4 and the ZMP values are plotted and shown in Figure 6.9

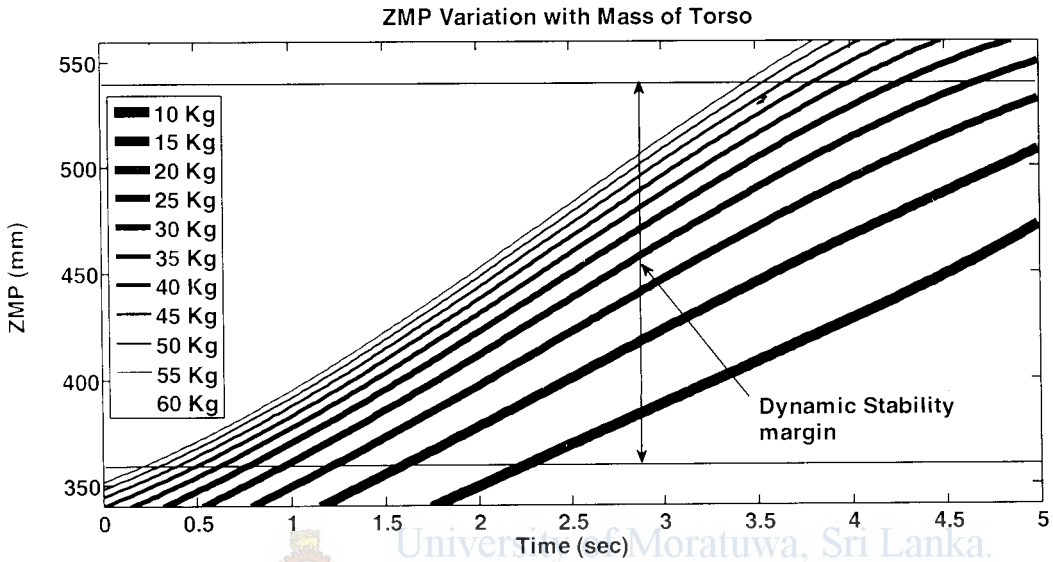


Figure 6.9-ZMP variation keeping torso angle at 15 deg. and varying torso mass

From Figure 6.8 it can be seen that by adding torso a weight dynamic stability can be improved. The Figure 6.9 shows the torso angle at 15 degrees and still it is possible to increase the stability by adding more weights to the torso. Also seen that by adding initial weights10-35Kg the dynamic stability has increased by much wider gap rather than at latter steps of weights 35-60 Kg. By observing this behavior it can be noted much improvement cannot be expected after some marginal weight.

6.5 Effect on length of torso on dynamic stability of robot

The effect of length of torso has on the dynamic stability of the bipedal robot is analyzed in this section.
The MATLAB program A.10 in appendix A has been used with data in Table 6.5 and the ZMP values are plotted and shown in Figure 6.10.

Table 6.5- Parameters for the calculation of ZMP of bipedal robot with different length of Torso

L_1, L_4	460 mm		x_1	100 mm
L_2, L_5	480 mm		x_2	450 mm
L_3, L_6	180 mm		x_3	800 mm
SL	700 mm		H	700 mm
g	9.81m/s^2		m_1, m_4	5.0 Kg
$x^{"}$	350 mm		m_2, m_3	5.2 Kg
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		α	5^0

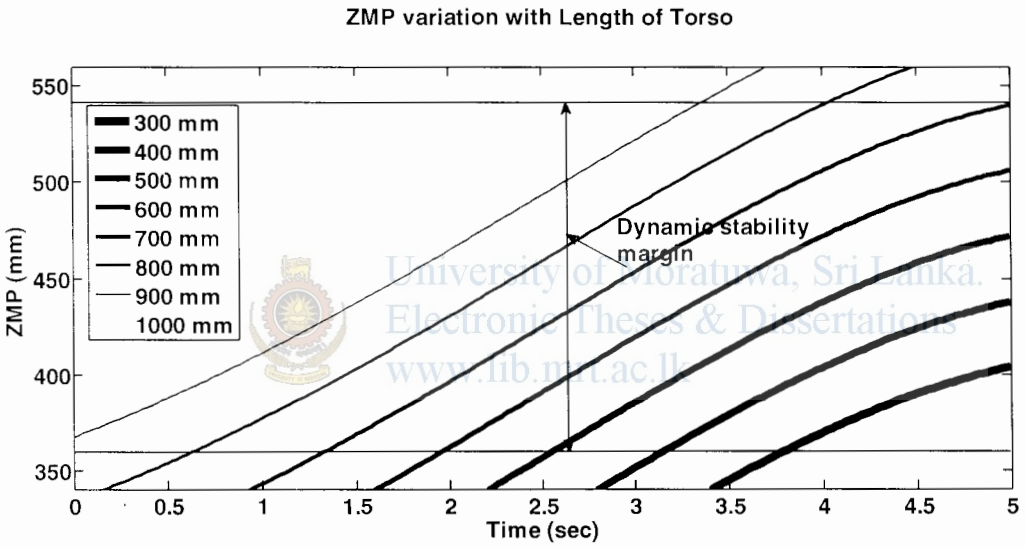


Figure 6.10-ZMP variation keeping torso angle at 10 deg. and varying torso length

In Figure 6.10 the variation of ZMP can be seen by increasing the length of torso. There is a greater increase in dynamic stability margin due to the increase in torso length. But there is a practical limitation.

6.6 Effect on weights of lower body on dynamic stability of robot

The effect of length of torso on the dynamic stability of the bipedal robot is analyzed in this test case. The MATLAB program A.10 in appendix A has been used with data in Table 6.6 and the ZMP values are plotted and shown in Figure 6.11.

Table 6.6- Parameters for the calculation of ZMP of bipedal robot with different weights of lower body

L_1, L_3	460 mm		x_1	100 mm
L_2, L_4	480 mm		x_2	450 mm
L_5	180 mm		x_3	800 mm
SL	700 mm		H	700 mm
g	9.81m/s^2		$x^{"}$	350 mm
T	5 sec		t_0	0 sec
t_1	1.25 sec		t_2	3.75 sec
t_3	5.00 sec		θ_T	10^0

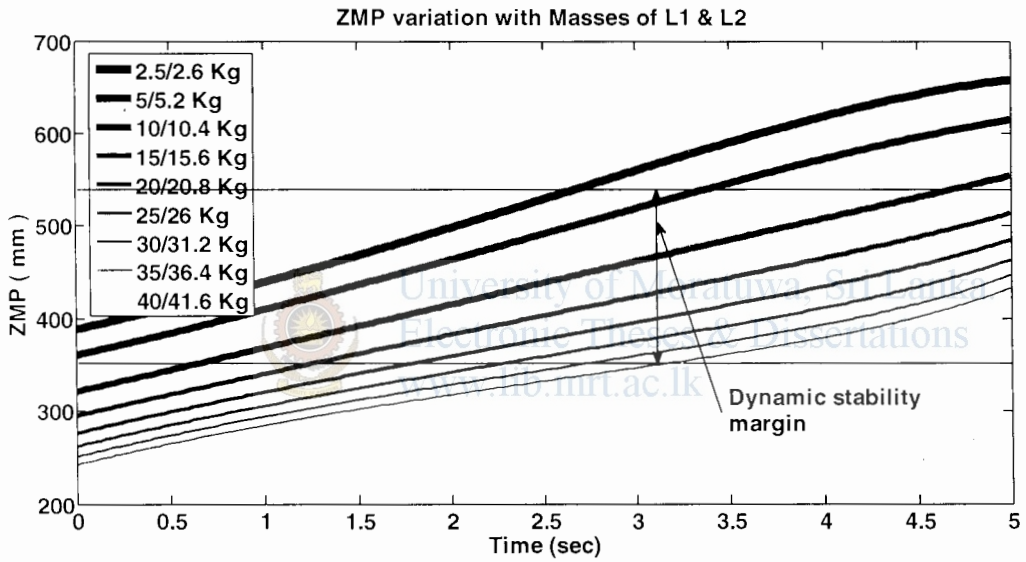


Figure 6.11-ZMP variation keeping torso angle at 10 deg. and varying the weights of lower body

The Figure 6.11 shows the behavior of ZMP when varying the weights of the legs L_1 & L_2 on slope angle of 5^0.The graph clearly shows the loss of dynamic stability with increase of weights of L_1 & L_2 .

6.7 Behavioral analysis of joint angles

The variation of joint angles angular position, angular velocity and angular acceleration are illustrated in figures 6.8, 6.9, 6.10 and 6.11.
The smooth variation ensures the smooth motion of the bipedal robot.

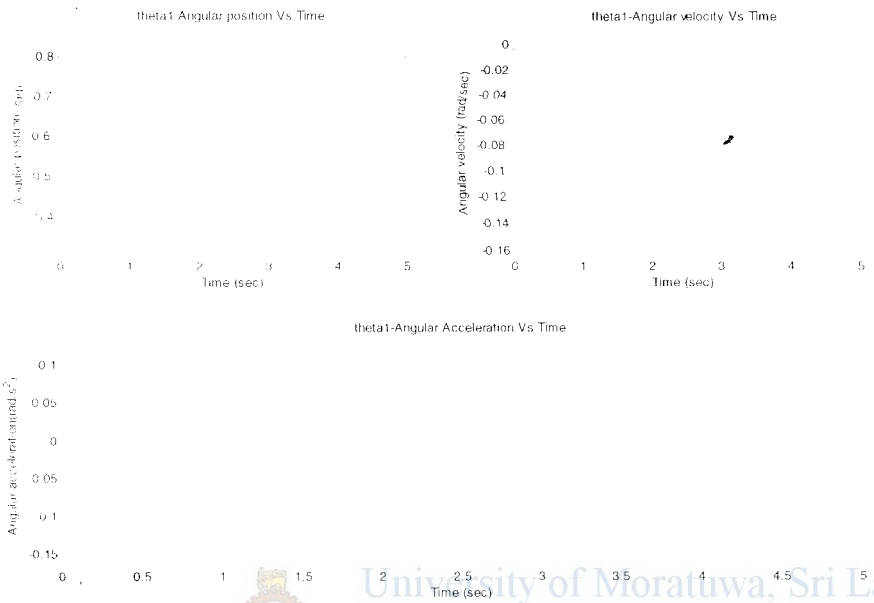


Figure 6.12: Angular position, Angular velocity and Angular acceleration of joint angle θ_1

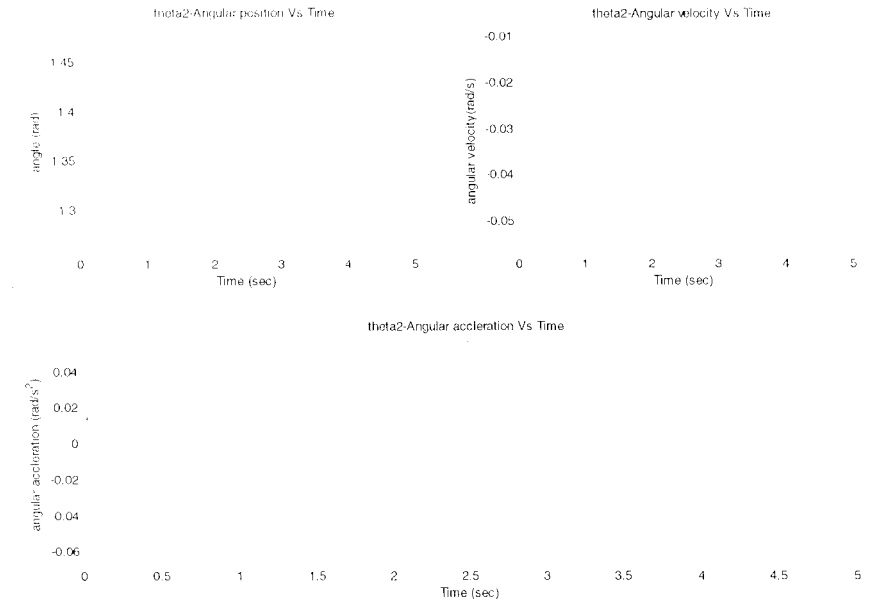


Figure 6.13: Angular position, Angular velocity and Angular acceleration of joint angle θ_2

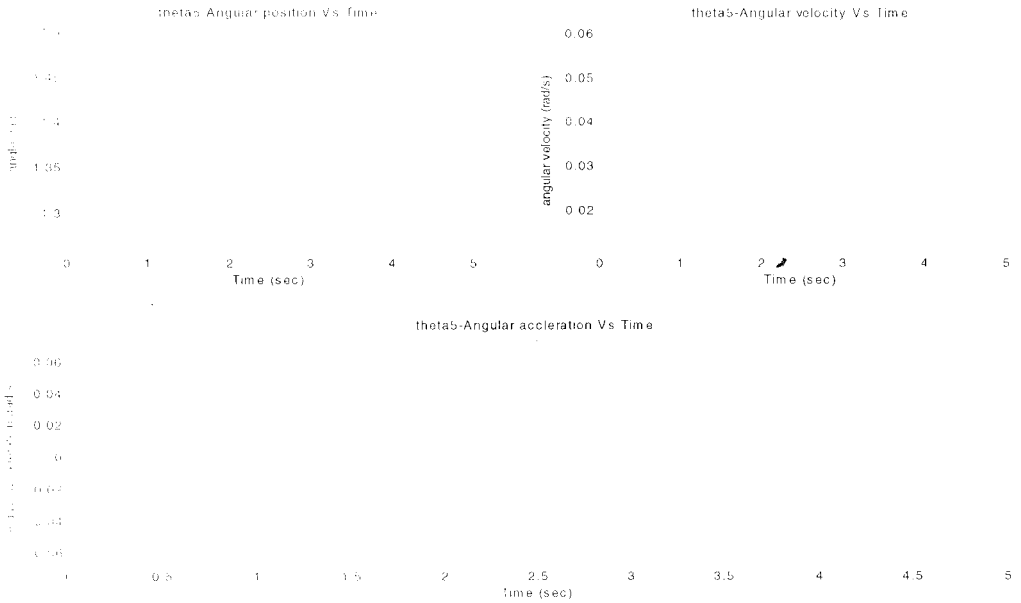


Figure 6.14: Angular position, Angular velocity and Angular acceleration of joint angle θ_5

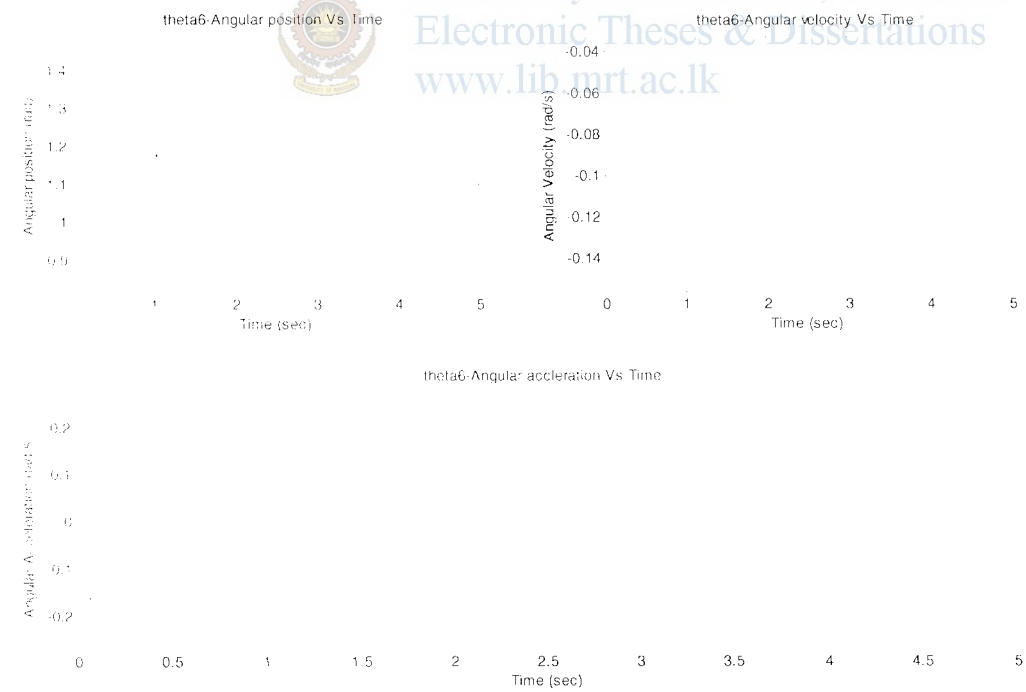


Figure 6.15: Angular position, Angular velocity and Angular acceleration of joint angle θ_6

References:

[1] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA: MIT Press, 1986.

[2] Miomir Vukobratovic and Branislav, "Zero-Moment Point-Thirty five years of its life," *International Journal of Humanoid Robotics*, vol.1, No. 1, pp.157-173, 2004.

[3] J. Yamaguchi,Y. Takanishi and I, Kato,"Development of a biped walking robot compensating for three axis movement by trunk motion,"*IEE/RSJ International workshop on intelligent robotics and systems,* vol. 2, pp. 561-566, 1999.

[4] A. Agrawal and S.K. Agrawal, "An approach to identify joint motions for dynamically stable walking," *ASME Trans. Mechanical Design,* vol.128, pp.649-653.P, 2006.

[5] Goswami, A., "Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots", *Robotics and Automation, 1999.Proceedings, 1999 IEEE Int. Conference on,* Volume: 1, 1999, pp. 47-52 vol.1.

[6] Honda Corporation. Say hello to asimo. http://asimo.honda.com, Accessed December 27, 2009.

[7] L. Geppert. Qrio, the robot that could. IEEE Spectrum, 41(5):34–37, May 2004.

[8] Massachusetts Institute of Technology (MIT) Leg laboratory ,http://www.ai.mit.edu/projects/leglab/robots/robots, Accessed September 21, 2009.

[9] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi. Humanoid robot HRP-2. In Proc IEEE Int Conf Robotics and Automation (ICRA'04), pages 1083–1090, April 2004.

[10] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue.Design and development of research platform for perception-action integration in humanoid robot : H6. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00), volume 3, pages 1559–1564, 2000.

[11] Fujitsu Automation. Humanoid robot.http://www.techjapan.com/Article1037.html, Accessed August 25, 2009.

[12] M. Gienger K. Loffler and F. Pfeiffer. Sensors and control concept of walking "Johnnie". Int J Robot Res, 22(3):229–239, March 2003.

[13] J.-H. Kim; J.-H. Oh. Realization of dynamic walking for the humanoid robot platform khr-1. Adv Robotics, 18(7):749–768, August 2004.

[16] R.R. Murphy. Trial by fire [rescue robots]. IEEE Robotics and Automation Magazine, 11(3):50-61, 2004.

[17] N. Robertson. Meet packbot: The newest recruit.http://archives.cnn.com/2002/TECH/science/08/01/packbot/. August 1, 2002. Accessed June 25, 2005.

[18] J. Bares and D. Wettergreen. Dante II: Technical description, results and lessons learned. IntJ Robot Res, 18(7):621–649, 1999.

[17] California Institute of Technology NASA Jet Propulsion Laboratory. Mars exploration rover mission. http://marsrovers.jpl.nasa.gov/home/, Accessed August 15, 2009.

[18] R.K.Mittal, and I.J.Nagrath ,Robotics and Control, Tata McGraw-Hill, 2003.

[19] John J. Craig, Introduction to Robotics:Mechanics and Control, Prentice Hall, 2004.

[20] RoboWorks –A tool for realtime interactive 3D modeling and animation with distributed simulation, http://www.newtonium.com, Accessed December 28, 2009.

[21] F.Plestan, W.G.Jessy, R.Westervelt, A.Gabrial, "Stable walking of a 7-DoF biped robot," IEEE Trans.Robot.Autom.,vol.19, pp. 653-668 , 2003.

[22] M.G.A.P Abeyratne, "Modelling of bipedal robot negotiating slopes" Master Thesis, University of Moratuwa, January 2010.

APPENDIX – A : MATLAB programs

Program A.1 : Program to calculate trajectory coefficients symbolically

```
x3 x1 x2 alpha c0 c1 c2 c3 c4 c5 c6 c7 f
*cos(alpha) x3^2*cos(alpha)^2 x3^3*cos(alpha)^3;1 x1*cos(alpha)
alpha)^2 x1^3*cos(alpha)^3;1 (x1*cos(alpha)+x2*cos(alpha))/2
(alpha)+x2*cos(alpha))^2/4 (x1*cos(alpha)+x2*cos(alpha))^3/8;1
(alpha)+x3*cos(alpha))/2  (x2*cos(alpha)+x3*cos(alpha))^2/4
alpha)+x3*cos(alpha))^3/8 ]
c2 c3]
in(alpha) ; x1*sin(alpha); x2*sin(alpha)+f/2 ;x3*sin(alpha)+f/2]
(R)
```

Program A.2 : Program to calculate trajectory coefficients numerically

```
tion [c0,c1,c2,c3]=coefficient(x1,x2,x3,alpha,f)
(x1^3*x3*sin(alpha)+x1^3*f-x1^3*x2*sin(alpha)-
*sin(alpha)*x3^2+3*x1^2*x3*x2*sin(alpha)-
*x3*f+x1^2*x2*f+x1^2*x2^2*sin(alpha)+x1*sin(alpha)*x3^3+2*x1*x3^2*f
alpha)*x2^2*x3-3*x1*x2^2*f-
alpha)*x2^2+3*x1*sin(alpha)*x3^2*x2-x3^3*f-
*x3^3*x2+sin(alpha)*x2^2*x3^2-
-3*x2^2*f*x3+2*sin(alpha)*x2^4-2*sin(alpha)*x2^3*x3)*x1*x3/(-
/(x1-x3)/(2*x1-x3-x2)/(x1-x2)/(x1-2*x3+x2);
-
*x3^3*sin(alpha)+4*x1*x3^4*sin(alpha)+3*x2^4*sin(alpha)*x3+x1^3*x2
alpha)-2*x3^2*x2^3*sin(alpha)+3*x3^3*x2^2*sin(alpha)-
*x2*sin(alpha)+5*x2^4*sin(alpha)*x1-
sin(alpha)*x1^2+2*x1^3*f*x2+2*x1^3*f*x3-3*x1^3*x3^2*sin(alpha)-
x1^2-6*x3*f*x2*x1^2+6*x3^2*f*x2*x1+6*x3^2*f*x2^2+2*x1^4*f-
-2*x3^3*f*x2-2*x3^3*f*x1-5*x3*x2^2*sin(alpha)*x1^2-
2*x2^2*sin(alpha)*x1+28*x3^2*x2*sin(alpha)*x1^2+14*x3^3*x2*sin(alph
*x1+2*x1^3*x3*sin(alpha)*x2)/(-x2+x3)/(x1-x3)/(2*x1-x3-x2)/(x1-x2)/(x1-
*x1-x2)/cos(alpha);
-*(-x1^3*x3*sin(alpha)+x1^3*x2*sin(alpha)+3*x1^3*f-
x1^2*sin(alpha)*x3^2+13*x1^2*x3*x2*sin(alpha)-5*x1^2*x2^2*sin(alpha)-
*x2*f-x1*sin(alpha)*x3^3-
sin(alpha)*x2^2*x3+2*x1*sin(alpha)*x2^3-
*x2^2*f+13*x1*sin(alpha)*x3^2*x2-
f*x1+sin(alpha)*x3^3*x2+3*x3^2*f*x2-
*sin(alpha)*x2^2*x3^2+3*x2^2*f*x3+2*sin(alpha)*x2^4+2*sin(alpha)*x2^3*x3
+x2-x3)/(x1-x3)/(2*x1-x3-x2)/(x1-x2)/(x1-2*x3+x2)/cos(alpha)^2;
-4*(x3+x1-2*x2)*(-x1*x3*sin(alpha)+x1*sin(alpha)*x2+x3*sin(alpha)*x2-
*sin(alpha)+x1*f-f*x3)/(-x2+x3)/(x1-x3)/(2*x1-x3-x2)/(x1-x2)/(x1-
*x1-x2)/cos(alpha)^3;
```

Program A.3 : Program to calculate joint angle θ_2

```
function [theta2 theta22 Ctheta2,Stheta2 ]=output_3(X,Y,L1,L2)
c2=((X^2+Y^2)-(L1^2+L2^2))/(2*L1*L2)
s2=sqrt(1-(Ctheta2^2))
theta2=atan2(Stheta2,Ctheta2)
theta22=180/pi*theta2
```

Program A.4 : Program to calculate joint angle θ_1

```
function [theta1  ]=output_5(X,Y,theta2)
k1=480+480*cos(pi*theta2/180);
k2=480*sin(pi*theta2/180);
theta11=atan2(Y,X)-atan2(k2,k1);
theta1=180*theta11/pi;
```

Program A.5 : Program to calculate joint angle θ_5

```
function [theta51=output_200(L1,L2,sx)
x=-150.*cosd(5);
y=150.*sind(5)+tand(5).*sx;
s=((X.^2+Y.^2)-(L1.^2+L2.^2));
s=s./L1.^L2;
s=s/4;
theta5=acosd(s)
```


Program A.6 : Program to calculate joint angle θ_6

```
function [theta6  ]=output_201(sx,theta5)
y=-150.*cosd(5);
x=150.*sind(5)+tand(5).*sx;
k1=-480.*cosd(theta5);
k2=480.*sind(theta5);
theta66=atan2(Y,X)-atan2(k2,k1);
theta6=(theta66*180)/pi;
```

Program A.7 : Program to calculate joint angle trajectory coefficients symbolically

```
function[eq1,eq2,eq3,eq4,eq5,eq6]=output_1003(thetas,thetae,theta1s,theta
1e,theta2s,theta2e,ts,te,t1,t2,a0,a1,a2,a3,a4,a5)
eq1=-thetas+a0+a1*ts+a2*ts^2+a3*ts^3+a4*ts^4+a5*ts^5;
eq2=-thetae+a0+a1*te+a2*te^2+a3*te^3+a4*te^4+a5*te^5;
eq3=-theta1s+a1+2*a2*ts+3*a3*ts^2+4*a4*ts^3+5*a5*ts^4;
eq4=-theta1e+a1+2*a2*te+3*a3*te^2+4*a4*te^3+5*a5*te^4;
eq5=-theta2s+2*a2+6*a3*t1+12*a4*t1^2+20*a5*t1^3;
eq6=-theta2e+2*a2+6*a3*t2+12*a4*t2^2+20*a5*t2^3;
[a0,a1,a2,a3,a4,a5]=solve(eq1,eq2,eq3,eq4,eq5,a0,a1,a2,a3,a4,a5)
```

Program A.8 : Program to calculate joint angle trajectory coefficients numerically

```
[a0,a1,a2,a3,a4,a5]= output_1004(thetas,thetae,theta1s,theta1e,the
ta2e,ts,te,t1,t2)

-40*te^2*t1^3*ts^2*t2*thetas-30*te^3*t1^3*ts^4*theta2e-
 4*t1^3*ts*t2*theta1s-
 8*t1^3*ts*t2*thetas+60*te^3*t1^3*ts^2*t2*theta1s+120*te^3*t1^3*t2^
theta1s+10*te^3*t1^3*ts^5*theta2e-
 *t1^3*ts^2*theta2e+30*te^4*t1^3*ts^3*theta2e+360*te^2*t1^3*t2^2*ts
 s-
*t1^3*t2^2*ts^2*theta1s+240*te^2*t1^3*ts^2*t2*thetae+60*te^2*t1^3
*t2*theta1s+20*t2^3*te^5*thetas-
2^2*ts^4*te^2*thetae+20*t2^3*ts^4*te^2*theta1e-
 2*ts^6*te*theta1e-36*t1*te^4*t2^2*ts^2*theta1e-
2*te^6*thetas+108*t2*ts^5*t1^2*thetae+24*t1^2*ts^5*te^3*theta2e+120
 1^2*t2^3*thetas+120*t2^2*ts^3*t1^3*thetae+24*t2^2*ts^6*thetae+180*
t2^2*te*thetae-360*t2^2*ts^2*t1^3*te*thetae-
 s^4*t2^2*te^2*theta1e+36*t2*ts^4*t1^2*te^2*theta1e-
 *te^5*ts^3*theta2e-108*t2*ts^5*t1^2*te*theta1e-
 *ts^4*t1^2*te*thetae-
 s^4*t1^2*te^2*theta1s+108*t1*ts^5*te*t2^2*theta1e-
 ts^6*thetae-20*t1^3*ts^5*thetae+24*t1*ts^4*te^5*theta2e+24*t1^2*t
 *theta1s-24*t1^2*ts^4*te^3*theta1e-
 theta1s*te^6*t2+108*t1*te^5*t2^2*thetas+240*te^3*t2^3*t1*ts*thetas
 *t2*te^4*theta1e-
 theta1s*t2*te^5+120*t2^2*ts^3*thetas*te^3-
2^2*ts^3*t1^3*te*theta1e-180*t1^2*ts^3*t2*te^2*thetae-
2*ts^2*t2*te^3*thetae+180*t1^2*ts^2*t2*te^3*thetas-
2*ts^3*t2*te^3*theta1s+36*t1^2*ts^3*t2*te^3*theta1e+180*t1^2*ts^3*t
2*thetas+30*t2^3*ts^4*te^3*theta2s-20*t2^3*ts^4*te*thetae-
2^2*ts^4*te^3*theta1s+24*t1^2*ts^5*te^2*theta1e+24*t1^2*ts^6*te*theta
*t1^2*ts^5*te*thetae-20*t2^3*ts^5*thetae-
3*ts^5*te*theta1e+20*t1^3*ts^4*te*thetae-
3*ts^4*te^2*theta1e+3*t1*ts^6*te^3*theta2e+120*t1^2*ts^4*te^2*theta
*t2^3*te^4*ts^2*theta1s-
ts^3*t1^2*t2^3*thetae+24*t2^2*te^4*ts^3*theta1e-
e^3*t1^2*t2^3*ts*theta1s+360*te*t1^2*t2^3*ts^2*thetae-
0^2*te^4*ts^3*theta1s-
2^3*te^4*ts^3*theta2s+20*t2^3*te^4*ts*thetas+36*t2*te^4*t1^2*ts^2*the
-36*t2*te^4*t1^2*ts^2*theta1s+3*t2*te^7*ts^2*theta2s-
2*te^5*t1^2*thetas+3*t2*te^6*theta2s*ts^3-
 s^4*t1*te*theta1e+20*t2^3*ts^5*te*theta1e-
*ts^5*te^2*theta1e+24*t2^2*ts^4*te^3*theta1e+240*ts^2*t2^3*t1*te^2
s-40*ts^2*t2^3*t1*te^2*thetae-
ts^5*t2^2*thetae+3*t1*ts^7*te^2*theta2e-
^3*t1^2*t2^3*ts^2*theta1e-
2*t1^2*t2^3*ts*thetas+120*te*t1^2*t2^3*ts^3*theta1e+120*te^2*t1^2*
*ts^2*theta1s-
t2^3*te^4*t1*thetas+108*t2*te^5*ts*t1^2*theta1s+180*t2*te^4*ts*t1^2*t
s+120*t2^2*te^4*ts^2*thetas+120*t2^3*ts^4*t1*thetae-
1^2*te^6*ts*theta1s+20*t1^3*te^4*ts^2*theta1s+120*t2^3*te^4*t1*ts*the
-180*t1*te^2*ts^3*thetas*t2^2-
1*ts^2*te^3*thetas*t2^2+180*t1*ts^2*te^3*thetae*t2^2-
1*ts^6*te^2*theta1e+6*t1*ts^6*te*thetae+6*t1*ts^5*te^2*thetae+40*t1^3*
*te^3*theta1e-160*t1^3*ts^3*te^2*thetae-
```

```
ts^3*te^3*theta1s+160*t1^3*theta5*ts^2*te^3+6*t1*ts^7*theta2e-
te^7*ts^2*theta2e-24*t1^2*te^4*ts^3*theta1e-20*t1^3*te^4*ts*theta5-
te^5*ts^2*theta1s-108*t1*te^5*t2^2*ts*theta1s-
                    theta2e-
          t1^3*t2^2*theta5+24*t2^2*te^5*ts^2*theta1s+24*t2^2*te^5*ts^3*the
                    theta1s+6*t2*theta5*te^7-
                    theta1s*te^3+4*ts^6*te^4*theta2e-
                    60*ts^2*t2^2*t1*te^2*theta1e-
                    t1*te*theta2e-
t1^3*t1*te^5*theta1s-60*ts^2*t2^3*t1*te^3*theta1e+120*ts^4*t1^3*t
theta1e+12*ts^2*t1^2*theta2e*te^6-
t1^2*theta2e*te^2+160*t2^3*ts^3*theta2e*te^2-
ts^3*theta2e*te^3-40*t2^3*ts^3*te^3*theta1s-
ts^3*te^3*theta1e+36*t1*ts^4*t2^2*te^2*theta1s-
t1^3*ts^3*t1*theta1e-2*ts^3*theta2s*te^7-
te^6*theta2s*ts^2+12*t2^2*te^2*theta2s*ts^6-24*t1^2*te^5*ts*theta
t1^3*te^5*ts*theta1s-20*t1^3*te^5*theta5-3*t1*te^6*ts^3*theta2e-
theta2e+120*te^4*t1^3*t2*theta5+240*ts^3*t1^3*te*t2*theta2e-
t1*te^4*t2^2*ts*theta5+36*t1*te^4*t2^2*ts^2*theta1s+120*te^2*t1^3*t2^
theta1e-
s^4*t1^3*t2*theta2e+180*t1*te^2*ts^3*theta2e*t2^2+36*t1*te^3*ts^3*thet
t2^2-36*t1*te^3*ts^3*theta1e*t2^2-
ts^4*te^3*theta2e+6*t1*ts^2*theta1s*te^6+6*t1*ts^3*theta1s*te^5+6*t1
ts*te^7+18*t1*ts^4*te^4*theta1e-90*t1*theta5*ts^3*te^4-
te^5*ts^2-6*t1*theta5*te^6*ts-6*t1*ts^7*te*theta1e-
t1^3*theta1e-18*t1*ts^4*te^4*theta1s+24*t2^2*ts^5*te*theta2e-
t1^3*ts^2*t2*theta1e-24*t2^2*ts^5*te^3*theta2s-
3*ts^5*te^2*theta2s+24*t2*ts^5*te^4*theta2s-3*t2*ts^6*theta2s*te^3-
te^3*t2^3*theta5+24*t1^2*te^4*ts^3*theta1s-
te^4*ts^2*theta5+10*t2^3*te^5*ts^2*theta2s+24*t2^2*te^6*ts*theta
t2*te^5*ts*theta5-
theta5*te^7+2*ts^3*theta2e*te^7+24*t1^2*te^6*theta5-
3*t2^3*t1*ts^3*theta1s+90*ts^4*t2*te^3*theta2e+6*ts^7*t2*te*theta1e+
t2*te^3*theta1e-120*t1^2*te^3*theta2e*ts^3-
12*te^3*theta5*ts^3-
theta5*ts^3*te^4+6*t2*theta5*te^6*ts+6*t2*theta5*te^5*ts^2-
6*te^4*theta2s+4*ts^4*theta2s*te^6+6*ts^6*t2*te^2*theta1e-
6*t2*te*theta2e-6*ts^5*t2*te^2*theta2e-
theta1s*te^7*t2+18*ts^4*t2*te^4*theta1s)/(180*ts^2*t1^2*te*t2^3-
te^7-3*ts^7*t2-10*ts^5*t2^3+12*t2^2*ts^6+10*te^5*t2^3-
3*t1^2*t2^3-54*t1^2*t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
t2^2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
4*t2*t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
4*te*t2^3+60*t2^2*ts^3*t1^3-
2*ts*te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
5*ts^5*t2+80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
te^2*ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
t1^2*te^2*t2^3+120*ts*t1*te^3*t2^3+90*t1*t2^2*te*ts^4+120*ts^3*te*
40*t1^2*t2*ts*te^4-120*t2*ts*te^3*t1^3-
ts^2*te*t1^3+180*t2^2*ts*te^2*t1^3-
te^6+3*te^7*t2+3*ts^7*t1-12*ts^6*t1^2+10*t1^3*ts^5-
t1^3*te^5+45*ts^3*t1*te^4-
5*te*t1^2+60*ts^4*te^2*t1^2+3*ts^6*t1*te-
s^4*t1*te^3+80*t1^3*ts^2*te^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
```

```
te^2*ts^3+12*t1^2*te^6-60*t1^2*ts^2*te^4-
6*ts-3*t1*te^2*ts^5-3*t1*te^5*ts^2-12*ts*t1^2*te^5);
(-120*t1^2*t2^3*te^3*theta1s+20*t2^3*ts^5*theta1e-
*ts^5*theta2s+5*ts^6*te^3*theta2e+24*te^4*t2^2*ts^2*theta1s+30*
*theta2e-30*ts^5*t2^2*te^2*theta2s-
te^4*ts*theta1s+30*te^5*t2^2*ts^2*theta2s-6*te^6*ts*t2*theta1s-
t1^3*theta1e-30*te^4*t2^2*ts^3*theta2s+36*te^4*ts^2*theta1e*t2^2-
ts^3*theta2e*t2^2-6*ts^6*te*t2*theta1e-24*ts^5*te*t2^2*theta1e-
te*t2*theta2s-20*t2^3*te^5*theta1s-
ts^2*theta2s+360*ts*t2^2*t1*te^3*theta2e-
t2^2*t1*te^3*theta1s+72*ts^4*t2^2*t1*te*theta1s+108*ts^4*t2^2*te*t1
2*ts^2*t2^2*te^3*t1*theta1s-72*ts^2*t2^2*te^3*t1*theta1e-
te^4*t1*theta1s+24*ts^6*theta2s*te*t2^2+3*te^7*ts^2*theta2e-
theta2e*t2^2-
t1^3*theta1s*t2^2-24*te^3*ts^4*theta1s*t2-
theta1e*t2*ts^3-36*te^3*ts^3*theta1e*t2^2-
4*ts^3*theta1s+40*te^2*t2^3*theta2s*ts^4+20*te^5*t2^3*theta2s*ts-
t2^3*ts^2*theta2s-6*ts^5*te^2*t2*theta1e+6*ts^4*te^3*t2*theta1e+2
2^2*theta1s-
5*ts^3*t2*theta2s+24*te^5*t2^2*ts*theta1s+6*te^7*t2*ts*theta2s+20*t
8*te^2*theta1e-6*te^5*ts^2*t2*theta1s+20*t2^3*ts^4*te*theta1e-
theta1s-
te^2*theta2e-24*ts^5*t1^2*te*theta1e+36*ts^3*t1^2*te^3*theta1s-
1^2*te^3*theta1e-
2*t1^2*te^4*theta1s+36*ts^4*t1^2*te^2*theta1s+24*ts^4*t1^2*te^2*the
-
8*t1^2*te^2*theta1s+180*ts^3*t1^2*te^2*theta2e+30*ts^3*t1*te^5*theta
ts^3*t2^2*t1*te^2*theta1s-
2^2*t1*te^2*theta1e-360*ts^3*t2^2*t1*te*theta2e-
2^2*t1*te^2*theta1s-
2^2*t1*te^2*theta1s+360*ts^2*t2^2*t1*te^2*theta2e-
2*t1*te^2*theta1s+72*ts^2*t1^2*t2*te^3*theta1e+120*ts^2*t1^2*t
theta1e-72*ts*t1^2*t2*te^4*theta1e+240*ts^2*t1^2*t2^3*te*theta1s-
1^2*t2*te*theta1e-30*ts^3*t1^2*te^4*theta2e-
t1^2*te*theta2e+30*ts^5*t1^2*te^2*theta2e-
t1^2*t2*te*theta1s-
1^2*t2^3*te^2*theta1e-360*ts*t1^2*t2*te^3*theta1s-
1^2*t2*te^3*theta2e+120*ts^3*t1^2*t2^3*theta1e-
1^2*te^5*theta2e-
6*te^2*theta2s+120*t1*t2^3*te^4*theta1s+180*ts^2*t1^2*te^3*theta1e
s^2*t1^2*te^3*theta1s-6*ts^2*t1*te^6*theta2e-
t2^3*ts^4*theta1e-
t1^3*te^4*theta1s+120*t2*ts^4*t1^3*theta1e-
^2*ts^3*t1^3*theta1e+120*te^3*theta2e*t2*ts^3-
e^3*theta1s*t2*ts^3-120*ts*t1^2*t2^3*te^2*theta1s-
1^2*t2^3*te*theta1s+720*ts*t1^2*t2^3*te*theta2e+108*ts*t1^2*t2*te^4
6*t2*ts^2*theta2s*te^6-
1^3*ts*te^2*theta1s+480*t2*t1^3*ts*te^2*theta2e-
3*te^3*ts*theta1s-
3*te^3*ts*theta1e+240*ts^2*te^2*t2^3*theta2e-
2*te^2*t2^3*theta1s+120*t2^2*t1^3*te^3*theta1s+6*ts^7*t2*theta1e-
6*t2^2*theta1e-30*ts^4*t2^2*te^3*theta2s-
t2^2*te^2*theta1e+180*te^2*ts^3*theta1s*t2^2+60*te^2*t2^3*ts^3*the
ts^2*t2^3*te^3*theta1s-60*ts^2*t2^3*te^3*theta1e-
ts^2*t1^3*te*theta1e+120*t2*ts^2*t1^3*te^2*theta1s-
3*theta2e+108*t1^2*te^5*t2*theta1s-
theta1s*te^5+6*t1*te^4*ts^3*theta1s-
```

```
...te^3*te^2*thetals-
...te^3*te^2*thetale+240*ts^2*t1^3*te^2*thetas-
...te^2*thetae+480*t1*t2^3*te^2*thetas-
...te^2*thetae+120*t2*ts^3*t1^3*te*thetals-120*t2*ts^3*t1^3
...e-480*t2*ts^2*t1^3*te*thetas+480*t2*ts^2*t1^3*te*thetae-
...te^3*te^2*thetale-240*t2^2*ts^2*t1^3*te*thetals-
...te^2*thetale+30*ts^4*t1^2*te^3*theta2e-
...te^5*theta2e+24*ts*t1^2*te^6*theta2e-
...t1^3*te*thetale+20*ts^2*t1^3*te^3*thetals+60*ts^2*t1^3*te^3*theta
...s*t1^3*te^4*thetals-
...t1^2*t2*thetale+20*ts^5*t1^3*te*theta2e+20*t1^3*te^5*thetals-
...te^6*thetals+6*t1*te^7*thetals-
...7*thetale+5*te^6*theta2s*ts^3+40*ts^2*t1^3*te^4*theta2e-
...t1^3*te^2*thetals-
...t1^3*te^2*thetale+240*ts^2*t1^3*te^2*thetas-
...t1^3*te^2*thetae-
...1*te^3*theta2e)/(180*ts^2*t1^2*te*t2^3-3*t1*te^7-3*ts^7*t2-
...t2^3+12*t2^2*ts^6-10*te^5*t2^3-60*ts^3*t1^2*t2^3-
...t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
...2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
...t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
...2^3+60*t2^2*ts^3*t1^3-
...te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
...t2-80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
...ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
...ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
...2*te^2*t2^3+120*ts*t1*te^3*t2^3+90*t1*t2^2*te*ts^4+120*ts^3*te*
...90*t1^2*t2*ts*te^4-120*t2*ts*te^3*t1^3-
...2*ts^2*te*t1^3+180*t2^2*ts*te^2*t1^3-
...te^6+3*te^7*t2+3*ts^7*t1-12*ts^6*t1^2+10*t1^3*ts^5-
...te^5+45*ts^3*t1*te^4-
...te*t1^2+60*ts^4*te^2*t1^2+3*ts^6*t1*te-
...4*t1*te^3+80*t1^3*ts^2*te^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
...3*te^2*ts^3+12*t1^2*te^6-60*t1^2*ts^2*te^4-
...te^6*ts+3*t1*te^2*ts^5-3*t1*te^5*ts^2+12*ts*t1^2*te^5);
...1/2*(3*theta2e*t1*te^7-
...2*ts^2*theta2s*te^4+12*t2^2*te^5*theta2s*ts+3*t2*te^2*theta2s*ts^5-
...3*theta2s*ts^4-3*t2*te^7*theta2s+80*ts^2*theta2s*te^3*t2^3-
...theta2s*ts^5-
...theta2s*t2^3+240*t2^2*t1^3*ts^2*thetale+120*t2^2*t1^3*ts^2*theta1
...1^3*ts*thetae-360*t2^2*t1^3*ts*thetas-
...1^3*te*thetas-360*t2^2*t1^3*te*thetae-
...t1^3*te^2*thetale+36*t2^2*t1*te^4*thetale-
```

```
t1^3*te^2*thetals-
t1*te^3*thetae+180*t2^2*t1*te^3*thetas-
t1*ts^3*thetae-180*t2^2*t1*ts^3*thetas-36*t2^2*t1*ts^4*thetals-
ts^4*t1*thetale+144*t2^2*te^4*t1*thetals+180*t1^2*t2*ts^3*thetae
t2^3*ts^2*thetals-360*t1^2*t2^3*ts*thetae-
t1^2*ts^3*thetas+36*t1^2*t2*ts^4*thetals+60*t1^2*ts^2*te^4*theta2e
ts^5*te*theta2e-36*t1^2*t2*te^4*thetale-
te^2*ts^4*theta2e+120*t2^2*t1^3*ts*te*thetals-
t2^2*t1^3*te*ts*thetale-
t2^2*ts^3*t1*te*thetals+36*t2^2*ts^3*te*t1*thetale-
t2^2*te^3*t1*ts*thetals+36*t2^2*te^3*ts*t1*thetale-
t2^2*t1*te^2*ts^2*thetals+36*t2^2*t1*te^2*ts^2*thetale-
t2^2*t1*ts^2*te*thetae+180*t2^2*t1*ts^2*te*thetas+180*t2^2*ts*t1*te^2
thetas-180*t2^2*ts*t1*te^2*thetae-
t1^3*ts^5*theta2e+10*t1^3*te^5*theta2e-
t1^2*te^6*theta2e+12*t1^2*ts^6*theta2e+36*t1^2*t2*ts*te^3*thetals-
t2*ts*te^3*thetale+120*t1^2*t2^3*ts*te*thetale-
t2^3*ts*te*thetals-
t2*t2*ts^3*te*thetale+144*t1^2*t2*ts^4*thetale-
s^4*te*theta2e+10*t1^3*te^4*ts*theta2e+120*t1^2*t2^3*te^2*thetal

t2*te^3*thetas+180*t1^2*t2*te^3*thetae+240*t1^2*t2^3*te^2*thetal
t1^2*t2^3*ts^2*thetale-
te^5*ts*theta2e+360*t1^2*t2^3*ts*thetas+360*t1^2*t2^3*te*thetas-
t2^3*te*thetae-144*t1^2*t2*te^4*thetals+60*t2^3*t1*ts^3*thetals-
t1^3*ts^2*thetas-240*t2^3*t1*te^2*thetas+240*t2^3*t1*te^2*thetae-
t1*te^3*thetals-
t1*te^3*thetale+240*t2^3*t1*ts^2*thetae+180*t2^3*t1*ts^3*thetale-
t1^2*ts^2*te^2*t2*thetale+180*t1^2*ts^2*te*t2*thetae-
t1^2*ts^2*te*t2*thetas-36*t1^2*ts^2*te^2*t2*thetals-
t1^2*ts*te^2*t2*thetas+180*t1^2*ts*te^2*t2*thetae+36*t1^2*t2*te*ts^3*
thetals+60*t2^3*t1*te^2*ts*thetals+240*t1^3*ts^2*t2*thetas-
t1^3*ts^2*t2*thetae-80*t1^3*ts^2*te^3*theta2e-
t1^3*ts^3*t2*thetals+80*t1^3*te^2*ts^3*theta2e+180*t1^3*te^3*t2*theta1
t1^3*te^2*t2*thetas-60*t2^3*t1*te*ts^2*thetals-
t1*te^2*ts*thetale-
t1*te*ts^2*thetale-240*t2^3*t1*ts*te*thetae-
t1^3*ts*te*thetas-60*t1^3*ts^2*te*t2*thetals-
te^2*ts*t2*thetals+240*t1^3*te*ts*t2*thetas-
t1^3*te*ts*t2*thetae-240*t1^3*te^2*t2*thetae-
t1^3*ts^3*t2*thetale+60*t1^3*te^3*t2*thetale-45*theta2e*ts^3*t1*te^4-
thetale*ts^6*t1*te+45*theta2e*ts^4*t1*te^3+3*theta2e*t1*te^6*ts-
te*thetale*t1*te^2*ts^5+3*theta2e*t1*te^5*ts^2+60*t1^3*te*ts^2*t2*thetale+
t1^3*s*te^2*t2*thetale-
thetale*ts^7*t1+3*t2*te*theta2s*ts^6+45*ts^3*theta2s*t2*te^4-
t2^2*ts^6*theta2s+10*ts^4*theta2s*te*t2^3-3*t2*te^5*theta2s*ts^2-
t2*te^6*theta2s*ts+60*t2^2*te^2*theta2s*ts^4+12*t2^2*te^6*theta2s-
te^4*theta2s*ts*t2^3-
ts^3*theta2s*te^2*t2^3+10*ts^5*theta2s*t2^3+3*ts^7*theta2s*t2)/(180*ts
t1^2*te*t2^3-3*t1*te^7-3*ts^7*t2-
ts^3*t2^3+12*t2^2*ts^6+10*te^5*t2^3-60*ts^3*t1^2*t2^3-
1^2*t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
t1*t2^2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
ts^4*t2*t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
ts^4*te*t2^3+60*t2^2*ts^3*t1^3-
t2^3*ts*te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
s^5*t2+80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
```

```
te^2*ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
t1*t2^2*ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
ts*t1^2*te^2*t2^3+120*ts*t1*te^3*t2^3+90*t1*t2^2*te*ts^4-120*ts^3*te*
t1^2*t2*ts*te^4-120*t2*ts*te^3*t1^3-
ts^2*te*t1^3+180*t2^2*ts*te^2*t1^3-
te^6+3*te^7*t2+3*ts^7*t1-12*ts^6*t1^2+10*t1^3*ts^5-
te^5-45*ts^3*t1*te^4-
t1^4-60*ts^4*te^2*t1^2+3*ts^6*t1*te-
te^3+80*t1^3*ts^2*te^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
te^2*ts^3-12*t1^2*te^6-60*t1^2*ts^2*te^4-
te^6*ts+3*t1*te^2*ts^5-3*t1*te^5*ts^2+12*ts*t1^2*te^5);
1/2*(-80*te^2*t2^3*theta1s+18*t1^2*ts^5*theta2e+60*te^2*t2^2*ts*theta1s-
te^2*t2^2*ts*theta1e-12*te^2*t2^2*ts^2*theta1s-
te^4*ts^3*theta2e+te^6*ts*theta2e+48*te^4*t2^2*theta1s+te^7*theta2e-12
*t2^2*ts^2*theta1e-20*t2^3*ts*te^2*theta1e+12*t2^2*ts^3*te*theta1e-
t2^3*ts^2*te*theta1e-20*t1^3*ts^3*theta1s-12*t1^2*te^4*theta1e-
te*theta2e+80*t2^3*te*ts*theta1e+18*t2^2*te^5*theta2s+theta2s*ts^7-
theta2s*t2^2*ts*te^4-
theta2s*ts^3*te*t2^3+40*theta2s*ts*te^3*t2^3+30*theta2s*t2^2*te*ts^4+2
theta2s*ts^4*t2^3-18*theta2s*t2^2*ts^5+theta2s*ts^6*te-
ta2s*te^6*ts+80*t1^3*ts^2*theta1s-20*t1^3*ts^4*theta2e-
t1^3*t2^2*theta1s+20*t1^3*te^4*theta2e+60*t1^3*te^3*theta1s+20*t1^3*te
theta1e-80*t1^3*ts^2*theta1e-60*t1^3*ts^3*theta1e+80*t1^3*te^2*theta1s-
t1^3*te^2*theta1e+240*t1^3*t2^2*theta1e-18*t1^2*te^5*theta2e-
t1^2*t2^3*theta1e+240*t1^2*t2^3*theta1s+60*t1^2*te^3*theta1e+48*t1^2*ts^
theta1e-
t1^2*ts^3*theta1s+12*t1^2*ts^4*theta1s+40*t1^3*ts^3*te*theta2e+20*t1^3*
e^2*theta1e-80*t1^3*ts*te*theta1e-80*t1^3*ts*te*theta1s-
ts*te^3*theta2e-
t1^3*t2^2*te*theta1s+120*t1^3*t2^2*ts*theta1s+120*t1^3*t2^2*ts*theta1
120*t1^3*t2^2*te*theta1e-20*t1^3*te^2*ts*theta1s-
t1^3*te*ts^2*theta1s+20*t1^3*te^2*ts*theta1e+60*t1^2*te^2*ts*theta1e-
t1^2*te^3*ts*theta1e+12*t1^2*ts^3*te*theta1s-
t1^2*ts^2*te*theta1s+120*t1^2*t2^3*te*theta1e+120*t1^2*t2^3*te*theta1s-
t1^2*ts^4*te*theta2e+30*t1^2*te^4*ts*theta2e-120*t1^2*ts*t2^3*theta1s-
3*t1^2*ts*t2^3*theta1e-
t1^2*te^4*theta1s-15*te^3*ts^4*theta2e+20*t2^3*ts^2*te*theta1s-
t2^2*ts^3*te*theta1s+80*te^2*t2^3*theta1e+80*t2^3*ts^2*theta1e+12*te^3*t
ts*theta1e-20*te^3*t2^3*theta1e-
2^2*ts^4*theta1e+60*t2^3*ts^3*theta1e-12*te^3*t2^2*ts*theta1s-
theta2e+60*te^3*t2^2*theta1s-
t2^2*ts^3*theta1e+60*t2^3*ts^2*theta1s-60*t2^2*ts^3*theta1e-
ts^4*theta1s-80*t2^3*ts^2*theta1s-
3*te*ts*theta1s+60*t2^2*ts^2*te*theta1s-
t2^2*ts^2*te*theta1e+15*ts^3*te^4*theta2s-15*ts^4*theta2s*te^3-
ts^3*t1^2*te*theta1e-60*ts^3*t1^2*theta1e+ts^5*te^2*theta2s+60*ts^2*t1^
e*theta1e+12*ts^2*t1^2*te^2*theta1s-12*ts^2*t1^2*te^2*theta1e-
theta1s*ts*te^2*t1^2+12*ts*theta1s*te^3*t1^2-60*theta1s*te^3*t1^2-
t1^2*t2^2s*te^5+12*te^4*t2^2*theta1e-te^7*theta2s-
ts^6*theta2e-te^5*ts^2*theta2e+60*t2^2*ts^3*theta1s+20*t2^3*ts*te^2*t
t1^2*te^4*theta2s*t2^3-60*t2^3*te^3*theta1s)/(180*ts^2*t1^2*te*t2^3-
te^7-3*ts^7*t2-10*ts^5*t2^3+12*t2^2*ts^6+10*te^5*t2^3-
ts^3*t1^2*t2^3-54*t1^2*t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
t1*t2^2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
ts^4*t2*t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
ts^4*te*t2^3+60*t2^2*ts^3*t1^3-
t2^2*ts*te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
```

```
ts^5*t2+80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
te^2*ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
t2^2*ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
2*te^2*t2^3+120*ts*t1*te^3*t2^3+90*t1*t2^2*te*ts^4+120*ts^3*te*
t1^2*t2*ts*te^4-120*t2*ts*te^3*t1^3-
t2*te*t1^3-180*t2^2*ts*te^2*t1^3-
t2*ts^7*t1-12*ts^6*t1^2-10*t1^3*ts^5-
te^6-4t*ts^3*t1*te^4-
te*t1^2+60*ts^4*te^2*t1^2+3*ts^6*t1*te-
4*t1*te^3+80*t1^3*ts^2*te^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
te^2*ts^3+12*t1^2*te^6-60*t1^2*ts^2*te^4-
*ts*3*t1*te^2*ts^5-3*t1*te^5*ts^2+12*ts*t1^2*te^5);
-30*t2*ts^3*theta1e+20*t2^3*ts^2*theta1s+2*ts^6*theta2s-
ts^3*theta2e-
^3*theta1s+6*t2*ts*te^3*theta1e+60*t2^3*ts*theta1e+30*t2*ts^3*th
2*ts^4*theta1s-20*t2^3*ts*te*theta1e-9*ts^5*t2*theta2s-
3*ts^2*theta1e-10*ts^2*te^4*theta2e-
te*theta2e+6*t2*te^4*theta1e+10*te^2*ts^4*theta2e+15*ts^4*t2*te*th
20*t2^3*ts*te*theta1s+6*t2*ts^3*te*theta1e+30*t1^3*te*ts^2*theta2e-
3*ts*te*theta1s-
3*ts*te^2*theta2e+60*t1^3*t2*ts*theta1e+20*t1^3*te*ts*theta1e+10*t1
3*theta2e-20*t1^3*ts^2*theta1s-60*t1^3*ts*theta1e+60*t1^3*ts*theta1s-
1^3*t2*theta1s+60*t1^3*te*theta1s-
3*te*theta1e+20*t1^3*te^2*theta1e+10*ts^3*theta2s*t2^3+60*t1^3*ts*t2
eta1s-60*t1^3*te*t2*theta1s+120*t1^3*t2*theta1e+6*ts^2*te^2*t2*theta1e-
2*theta2s*te*t2^3-30*ts^2*te*t2*theta1e+30*ts^2*te*t2*theta1s-
te^2*t2*theta1s-15*ts*t2*te^4*theta2s+30*ts*te^2*t2*theta1s-
e^3*t2*theta1e-60*t1^3*t2*te*theta1e-
4*theta1e+40*t1^3*te^2*theta1s+9*t1*ts^5*theta2e-
5*theta2e-
2^3*theta1e+120*t1*t2^3*theta1s+30*t1*te^3*theta1e-
e^3*theta1s+30*t1*ts^3*theta1e-
ts^3*theta1s+6*t1*ts^4*theta1s+2*te^6*theta2e+30*ts*theta2s*te^2*t2^
3*t1*te*theta1s-
4*te^2*theta2s+24*ts^4*t1*theta1e+2*ts^5*te*theta2s-
3*te*t1*theta1e+6*te^3*t1*ts*theta1s-6*te^3*ts*t1*theta1e-
5*ts*theta2s-24*te^4*t1*theta1s+10*te^4*ts^2*theta2s-
2*ts^4*theta1e-
6*theta2e+60*t1*t2^3*te*theta1e+60*t1*t2^3*te*theta1s-
1*ts^4*te*theta2e+15*t1*te^4*ts*theta2e-60*t1*ts*t2^3*theta1s-
1*ts*t2^3*theta1e-6*t2*te*ts^3*theta1s-
2^3*te^2*theta1e+30*t2*te^3*theta1s-
t2*te^3*theta1e+6*t1*te^2*ts^2*theta1s-
1*te^2*ts^2*theta1e+30*t1*ts^2*te*theta1e-30*t1*ts^2*te*theta1s-
s*t1*te^2*theta1s+30*ts*t1*te^2*theta1e-2*theta2s*te^6-
2*te^2*theta1s+40*t2^3*ts^2*theta1e+2*te^5*ts*theta2e-
1s*theta1s+9*t2*te^5*theta2s-10*theta2s*te^3*t2^3-
5*te*theta1s+60*t2^3*te*theta1e+24*t2*te^4*theta1s)/(180*ts^2*t1^2*te
-3*t1*te^7-3*ts^7*t2-10*ts^5*t2^3+12*t2^2*ts^6+10*te^5*t2^3-
3*t1^2*t2^3-54*t1^2*t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
1*t2^2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
4*t2*t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
4*te*t2^3+60*t2^2*ts^3*t1^3-
2*ts*te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
2*ts^5*t2+80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
2*te^2*ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
2*ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
```

```
...*te^2*t2^3+120*ts*t1*te^3*t2^3+90*t1*t2^2*to*ts^4+120*ts^3*to*
...-90*t1^2*t2*ts*to^4-120*t2*ts*te^3*t1^3-
...*ts^2*te*t1^3+180*t2^2*ts*to^2*t1^3-
...te^6+3*te^7*t2+3*ts^7*t1-12*ts^6*t1^2+10*t1^3*ts^5-
...+45*ts^3*t1*to^4-
...5*to*t1^2+60*ts^4*te^2*t1^2+3*ts^6*t1*te-
...4*t1*te^3+80*t1^3*ts^2*to^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
...3*te^2*ts^3+12*t1^2*te^6-60*t1^2*ts^2*te^4-
...6*ts+3*t1*te^2*ts^5-3*t1*te^5*ts^2+12*ts*t1^2*to^5);
...6*t1*ts^3*theta1s-24*t1*ts^2*thetas-
...thetas+24*t1*te^2*thetae-
...theta2e+6*ts^4*t1*theta2e-
...thetas-72*t1^2*t2*thetae+72*t1*t2^2*thetas-
...thetae-6*te^3*t1^2*theta2e-6*te^4*t1*theta2e-
...theta2s-ts^4*te*theta2e+24*ts^2*t2*thetas-
...t1^2*te^2*theta2e-
...3*t1*te*theta2e+12*ts*t1*te^3*theta2e+18*ts^2*t1^2*te*theta2e+36*t1
...ts*theta1s-36*t1^2*t2*te*theta1s+36*t1*t2^2*te*theta1s-
...2*ts*theta1s-ts^5*theta2e+te^5*theta2e-
...te*t2*theta1s+12*ts^3*te*t2*theta2s+12*t2^2*ts*te*theta1s-
...*te^3*theta2s+18*t2^2*ts*te^2*theta2s-18*t2^2*ts^2*te*theta2s-
...ts*t2*theta1s+24*te*ts*t2*thetas-24*te*ts*t2*thetae-
...t2*thetae-8*ts^2*te^3*theta2e-
...t2*theta1s+6*t2^2*ts^3*theta2s-
...2*ts*thetas+36*t2^2*ts*thetae+12*t2^2*ts^2*theta1s+8*te^2*ts^3*thet
...e^4*ts*theta2e-24*t2^2*te^2*theta1s+6*t2*te^4*theta2s-
...te^3*theta2s-
...2*te*thetas+36*t2^2*te*thetae+18*te^3*t2*theta1s+24*te^2*t2*thetas-
...te*t2*thetae-18*ts^3*t2*thetae-24*t2^2*ts^2*theta1e-
...e^2*theta1e-
...2*ts*theta1e-36*t1^2*t2*ts*theta1e+36*t1*t2^2*te*theta1e-
...t2*te*theta1e+6*te*ts^2*t2*theta1e-
...2*ts*theta1e+6*ts*te^2*t2*theta1e+6*te^3*t2*theta1e-
...theta2s+ts^5*theta2s-te*theta2s*ts^4-8*te^2*theta2s*ts^3-
...theta2s*ts+8*ts^2*theta2s*te^3+6*t1*te^2*ts*theta1s-
...ts^2*theta1e-12*t1^2*ts^2*theta1s-18*t1*te^3*theta1s-
...*theta1e+24*t1*ts^2*thetae-18*t1*ts^3*theta1e-
...theta1e+36*ts*t1^2*thetas+6*t1*te*ts^2*theta1s-
...2*ts*theta1e-6*t1*te*ts^2*theta1e+24*t1*ts*te*thetae-
...ts*te*thetas+12*ts*t1^2*te*theta1e-
...ts*t1^2*te*theta1s+36*t1^2*te*thetas-
...t1^2*te*thetae+24*t1^2*te^2*theta1s+12*t1^2*te^2*theta1e)/(180*ts^2*t1
...te*t2^3-3*t1*te^7-3*ts^7*t2-10*ts^5*t2^3+12*t2^2*ts^6+10*te^5*t2^3-
...*t1^2*t2^3-54*t1^2*t2*te^5+60*ts^4*t1*t2^3+54*t1^2*t2*ts^5-
...t1*t2^2*ts^5+54*t1*t2^2*te^5+60*te^3*t1^2*t2^3-60*te^4*t1*t2^3-
...ts^4*t2*t1^3+3*ts^2*t2*te^5-80*ts^2*te^3*t2^3-3*ts^6*te*t2-
...ts^4*te*t2^3+60*t2^2*ts^3*t1^3-
...t2^2*ts*te^5+12*t2^2*ts^5*te+3*te^6*ts*t2+10*te^4*ts*t2^3-
...e^2*ts^5*t2+80*te^2*ts^3*t2^3+60*t2*te^4*t1^3-60*t2^2*te^3*t1^3-
...t2^2*te^2*ts^4+45*te^3*t2*ts^4-45*ts^3*t2*te^4+60*t2^2*ts^2*te^4-
...t1*t2^2*ts*te^4-120*ts^3*t1*te*t2^3-90*t1^2*t2*te*ts^4-
...ts*t1^2*te^2*t2^3-120*ts*t1*te^3*t2^3-90*t1*t2^2*te*ts^4+120*ts^3*te*
...t1^3+90*t1^2*t2*ts*te^4-120*t2*ts*te^3*t1^3-
...ts^2*te*t1^3+180*t2^2*ts*te^2*t1^3-
...te^6+3*te^7*t2-3*ts^7*t1-12*ts^6*t1^2+10*t1^3*ts^5-
...*te^5+45*ts^3*t1*to^4-
...ts^5*te*t1^2-60*ts^4*te^2*t1^2+3*ts^6*t1*te-
```

```
...*t1*L2^3+80*t1^3*ts^2*te^3-10*t1^3*ts*te^4+10*t1^3*te*ts^4-
...*t1^3-10*t1^2*ts^6-60*t1^2*ts^2*te^4-
...*ts^2*t1*te^2*ts^3-3*t1*te^5*ts^2+12*ts*t1^2*te^5);
```

Program A.9 : Program to calculate ZMP of lower body

```
function
...output_10101(t,L1,L2,sx,theta1,theta2,theta5,theta6,q1,q2,q3,q4,m1,m2
,...,a,b,y,
...*L1*cos(theta1)+0.1*cos(pi/36);
...*L1.*sin(theta1)+y-L2.*sin(pi-(theta1+theta2));
...*L1.*cos(pi-(theta1+theta2));
y=0.5*L2*sin (pi-(theta1-theta2));
...*L2*cos(theta6)-0.45*cos(pi/36);
...*L2*sin(theta6)-0.45*sin(pi/36);
xx=0.5*L1*cos(pi-(theta5+theta6))+0.1*cos(pi/36);
...*sin(pi/36)-L2*sin(theta6)+0.5*L1*sin(pi-(theta5-theta6));
...0.006*t+.2034*t.^2-0.062*t.^3+0.006*t.^4;
...006+0.4068*t-0.186*t.^2+.024*t.^3;
...126*t-0.0951*t.^2-0.0292*t.^3+2.90685e-3*t.^4;
...126+0.1902*t-0.0876*t.^2+.01162/4*t.^3;
...=-0.3854*t+0.3258*t.^2-.0996*t.^3+.01*t.^4;
...-0.3854+.6516*t-.2988*t.^2+.04*t.^3;
...25*t-0.1056*t.^2+.0324*t.^3-3.2265e-3*t.^4;
...0.125-0.2112*t+.0972*t.^2-1.2906e-2*t.^3;
...*cos(q4)+4.*q23+L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3-q4);
...*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
...*L1.*q14-g.*sin(q3-q4);
...*(q1)-(b-g).*sin(q1)-0.5.*L1.*q11.^2;
...*sin(q1)-(b-g).*cos(q1)+0.5.*L1.*q22;
...*cos(q2)-g.*sin(q2)-0.5.*L2.*(q11+q12).^2;
...*sin(q2)+g.*cos(q2)-0.5.*L2.*(q21+q22);
...*sin(q3)-0.5.*L2.*q13.^2;
...*cos(q3)-0.5.*L2.*q23.^2;
...^2/12;
...^2/12;
...;
...;
...0.5.*(q13+q14).^2.*L1;
...0.5.*(q23+q24).*L1;
...*q21+m1.*x1.*(b-g)-m1.*a.*y1;
...*q22+m2.*x2.*(d-g)-m1.*c.*y2;
...*q13+m3.*x3.*(f-g)-m3.*e.*y3;
...*q24+m4.*x4.*(k-g)-m4.*j.*y4;
...*(b-g);
...*(d-g);
...*(f-g);
...*(k-g);
...s1/(u+v+w+rr);
```

Program A.10 : Program to calculate ZMP at 5degrees

```
function
  [output 10]03(t,L1,L2,L5,sx,sy,theta1,theta2,theta5,theta6,theta7,q1,q
,q7,q17,q27,m1,m2,m3,m4,m5,g,x,y)

  x=-0.5*L1*cos(theta1)+0.1*cos(pi/36);
   .5.*L1.*sin(theta1)+y+L2.*sin(pi-(theta1+theta2));
 x+0.5*L2*cos(pi-(theta1+theta2));
 y+0.5*L2*sin (pi-(theta1+theta2));
 .5*L2*cos(theta6)+0.45*cos(pi/36);
 .5*L2*sin(theta6)+0.45*sin(pi/36);
sx+0.5*L1*cos(pi-(theta5+theta6))+0.1*cos(pi/36);
.450*sin(pi/36)+L2*sin(theta6)+0.5*L1*sin(pi-(theta5+theta6));
.100*cos(pi/36)-sx+0.5*L5*cos(theta5+theta6-theta7);
.700+sy+0.5*L5*sin(theta5+theta6-theta7);
=-0.2406*t+.2034*t.^2-0.062*t.^3+0.006*t.^4;
 -.2406+0.4068*t-0.186*t.^2+.024*t.^3;
  -0.1126*t-0.0951*t.^2-0.0292*t.^3+2.90685e-3*t.^4;
  -.1126-0.1902*t-0.0876*t.^2+.0116274*t.^3;
  -.3384*t+0.3258*t.^2-.0996*t.^3+.01*t.^4;
  -.3384+.6516*t-.2988*t.^2+.04*t.^3;
  ...+t-0.1056*t.^2+.0324*t.^3-3.2265e-3*t.^4;
   .1+-0.2112*t+.0972*t.^2-1.2906e-2*t.^3;
 .*cos(q4)+4.*q23+L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3+q4);
 .*q23.*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
.*L1.*q13.*q14-g.*sin(q3+q4);
.*cos(q1)+(h-g).*sin(q1)-0.5.*L1.*q11.^2;
.*sin(q1)+(h-g).*cos(q1)+0.5.*L1.*q22;
.*cos(q2)+b.*sin(q2)-0.5.*L2.*(q11+q12).^2;
.*sin(q2)-b.*cos(q2)-0.5.*L2.*(q21+q22);
-.*sin(q3)-0.5.*L2.*q13.^2;
-.*cos(q3)+0.5.*L2.*q23.^2;
m1.*L1.^2/12;
+m2.*L2.^2/12;
 L1;
 L2;
  .*L5.^2/12;
  +.5.*(q13+q14).^2.*L1;
 - .*.*(q23+q24).*L1;
  .*cos(q7)-h.*sin(q7)-0.5.*L5.*(q13+q14+q17).^2;
 .*sin(q7)+h.*cos(q7)+0.5.*L5.*(q23+q24+q27);
  .*q21+m1.*x1.*(b-g)-m1.*a.*y1;
   .*q22+m2.*x2.*(d-g)-m2.*c.*y2;
  .*q23-m3.*x3.*(f-g)-m3.*e.*y3;
 .*q24-m4.*x4.*(k-g)-m4.*j.*y4;
  .*q27+m5.*x5.*(v6-g)-m5.*v5.*y5;
 1.*(b-g);
 2.*(d-g);
-3.*(f-g);
 m4.*(k-g);
 m5.*(v6-g);
  p+r+s+v55]/(u+v+w+rr+v66);
```

Program A.11 : Program to calculate ZMP at 10 degrees

```
function [x,y]=ZMP108(t,L1,L2,L5,sx,sy,q1,q2,q3,q4,q7,q17,q27,m1,m2,m3,m4,m5,g
);

x=sx+0.5.*L1*cos(q1)+0.1.*cos(pi/18);
y=sy+0.5.*L1.*sin(q1)+y+L2.*sin(pi-(q1+q2));
z=sz+0.5*L2*cos(pi-(q1+q2));
y=y+0.5*L2*sin (pi-(q1+q2));
x=sx+0.5*L2*cos(q3)+0.45*cos(pi/18);
y=sy+0.5*L2*sin(q3)+0.45*sin(pi/18);
x=sx+0.5*L1*cos(pi-(q4-q3))+0.1*cos(pi/18);
y=0.450*sin(pi/18)+L2*sin(q3)+0.5*L1*sin(pi-(q4+q3));
z=0.100*cos(pi/18)+sx+0.5*L5*cos(q4+q3-q7);
y=0.700+sy+0.5*L5*sin(q4+q3-q7);
q11=-0.2988*t+0.2385*t.^2-0.0688*t.^3+6.5e-3*t.^4;
q21=-.2988+0.477*t-0.2064*t.^2+2.6e-2*t.^3;
q12=-.0306*t+0.0243*t.^2-7.2e-3*t.^3+6.8065e-4*t.^4;
q22=-0.0306+0.0486*t-2.16e-2*t.^2+2.7226e-3*t.^3;
q13=-.3054*t+0.2439*t.^2-.0704*t.^3+7e-3*t.^4;
q23=-0.3054+.4878*t-.2112*t.^2+2.8e-2*t.^3;
q14=-.0952*t+.0759*t.^2-.022*t.^3+2.1161e-3*t.^4;
q24=-.0952+.1518*t-.066*t.^2+8.4644e-3*t.^3;
x=L1.*q23.*cos(q4)+4.*q23-L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3+q4);
y=L2.*q23.*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
L1.*q13.*q14-g.*sin(q3+q4);
a=a.*cos(q1)+(h-g).*sin(q1)-0.5.*L1.*q11.^2;
a=a.*sin(q1)+(h-g).*cos(q1)+0.5.*L1.*q22;
b=b.*cos(q2)+b.*sin(q2)-0.5.*L2.*(q11+q12).^2;
a=-a.*sin(q2)+b.*cos(q2)+0.5.*L2.*(q21+q22);
c=-c.*sin(q3)-0.5.*L2.*q13.^2;
e=-g.*cos(q3)+0.5.*L2.*q23.^2;
i=m1.*L1.^2/12;
j=m2.*L2.^2/12;
g=L1;
k=L2;
n=m5.*L5.^2/12;
m=+0.5.*(q13+q14).^2.*L1;
q=n-0.5.*(q23+q24).*L1;
b=n.*cos(q7)-h.*sin(q7)-0.5.*L5.*(q13+q14+q17).^2;
v=n.*sin(q7)+h.*cos(q7)+0.5.*L5.*(q23+q24+q27);
f=i.*q21+m1.*x1.*(b-g)-m1.*a.*y1;
f=j.*q22+m2.*x2.*(d-g)-m1.*c.*y2;
f=k.*q23+m3.*x3.*(f-g)-m3.*e.*y3;
f=4.*q24+m4.*x4.*(k-g)-m4.*j.*y4;
f=-n.*q27+m5.*x5.*(v6-g)-m5.*v5.*y5;
m1.*(b-g);
m2.*(d-g);
m3.*(f-g);
m=m4.*(k-g);
v=m5.*(v6-g);
[1+5+1+s+v55j/(u+v+w+rr+v66);
```

Program A.12 : Program to calculate ZMP at 15 degrees

```
[x,y]=output_10109(t,L1,L2,L5,sx,sy,q1,q2,q3,q4,q7,q17,q27,m1,m2,m3,m4,m5,g
,a,y)

sx=0.5.*L1*cos(q1)+0.1.*cos(pi/12);
sy=0.5.*L1.*sin(q1)+y+L2.*sin(pi-(q1+q2));
sx=sx+L2*cos(pi-(q1-q2));
sy=sy+L2*sin (pi-(q1+q2));
sx=sx+L2*cos(q3)+0.45*cos(pi/12);
sy=sy+L2*sin(q3)+0.45*sin(pi/12);
sx=sx-0.5*L1*cos(pi-(q4+q3))+0.1*cos(pi/12);
sy=sy+.480*sin(pi/12)+L2*sin(q3)-0.5*L1*sin(pi-(q4+q3));
sx=0.100*cos(pi/12)-sx-0.5*L5*cos(q4+q3-q7);
sy=0.700+sy+0.5*L5*sin(q4-q3-q7);
q1=-.33*t+0.2673*t.^2-0.0788*t.^3+7.5e-3*t.^4;
q2=-.33+0.5346*t-0.2364*t.^2+3e-2*t.^3;
q13=7.8e-3*t-6.3e-3*t.^2+1.8881e-3*t.^3-1.8518e-4*t.^4;
q23=7.8e-3-1.26e-2*t+5.6643e-3*t.^2-7.4072e-4*t.^3;
q14=-.2762*t+0.2238*t.^2-.066*t.^3+6.5e-3*t.^4;
q24=-0.2762-.4476*t-.198*t.^2+2.6e-2*t.^3;
q17=-.0848*t+.0687*t.^2-.0204*t.^3+1.9895e-3*t.^4;
q27=-.0848+.1374*t-.0612*t.^2+7.958e-3*t.^3;
x=L2.*q23.*cos(q4)+4.*q23+L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3+q4);
y=L2.*q23.*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
.*L1.*q13.*q14-g.*sin(q3+q4);
a=L.*cos(q1)+(h-g).*sin(q1)-0.5.*L1.*q11.^2;
b=n.*sin(q1)+(h-g).*cos(q1)+0.5.*L1.*q22;
c=b.*cos(q2)+b.*sin(q2)-0.5.*L2.*(q11+q12).^2;
d=-a.*sin(q2)+b.*cos(q2)+0.5.*L2.*(q21+q22);
e=-a.*sin(q3)-0.5.*L2.*q13.^2;
f=-a.*cos(q3)+0.5.*L2.*q23.^2;
x1=m1.*L1.^2/12;
x2=m2.*L2.^2/12;
x3=L1;
x4=L2;
x5=m5.*L5.^2/12;
v6=-0.5.*(q13+q14).^2.*L1;
v66=-0.5.*(q23+q24).*L1;
w=n.*cos(q7)-h.*sin(q7)-0.5.*L5.*(q13+q14+q17).^2;
rr=n.*sin(q7)+h.*cos(q7)+0.5.*L5.*(q23+q24+q27);
u=L1.*q21-m1.*x1.*(b-g)-m1.*a.*y1;
v=L2.*q22+m2.*x2.*(d-g)-m1.*c.*y2;
w=L3.*q23+m3.*x3.*(f-g)-m3.*e.*y3;
rr=L4.*q24-m4.*x4.*(k-g)-m4.*j.*y4;
v66=L5.*q27+m5.*x5.*(v6-g)-m5.*v5.*y5;
y1=m1.*(b-g);
y2=m2.*(d-g);
y3=m3.*(f-g);
y4=m4.*(k-g);
y5=m5.*(v6-g);
x=(u+v+w+rr+v66)/(u+v+w+rr+v66);
```

Program A.13 : Program to calculate ZMP at half step

```
function
  output_10107(t,L1,L2,L5,sx,sy,theta1,theta2,theta5,theta6,theta7,q1,q
  ,q5,q7,q17,q27,m1,m2,m3,m4,m5,g,x,y)

   .5*L1*cos(theta1)+0.1*cos(pi/36);
   .*L1.*sin(theta1)+y-L2.*sin(pi-(theta1+theta2));
  x=.5*L2*cos(pi-(theta1+theta2));
  y=0.5*L2*sin (pi-(theta1+theta2));
   .*L6*cos(theta6)-0.25*cos(pi/36);
   .*L2*sin(theta6)+0.25*sin(pi/36);
   x=.5*L1*cos(pi-(theta5+theta6))+0.1*cos(pi/36);
   .*sin(pi/36)+L2*sin(theta6)+0.5*L1*sin(pi-(theta5+theta6));
   .*cos(pi/36)+sx-0.5*L5*cos(theta5+theta6-theta7);
   .*sy-0.5*L5*sin(theta5+theta6-theta7);
  -.*166*t+1.032*t.^2-.5884*t.^3+0.112*t.^4;
  -.*6866+2.064*t-1.7652*t.^2-.448*t.^3;
   .*8866e-4*t-1.4947e-3*t.^2-8.5212e-4*t.^3-1.6198e-4*t.^4;
   .*8662e-4-2.9894e-3*t+2.5563e-3*t.^2-6.4792e-4*t.^3;
   .*t-0.047*t.^2+.0272*t.^3-.005*t.^4;
  .*s=-.0954*t-.0816*t.^2-.02*t.^3;
  -.*866*t+1.0956*t.^2-.6248*t.^3+.1185*t.^4;
  -.*8866-2.1912*t-1.8744*t.^2+.474*t.^3;
  .*q23.*cos(q4)-4.*q23+L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3+q4);
  .*q23.*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
  1.*q13.*q14-g.*sin(q3+q4);
  .*cos(q1)+(h-g).*sin(q1)-0.5.*L1.*q11.^2;
  .*sin(q1)+(h-g).*cos(q1)+0.5.*L1.*q22;
  .*cos(q2)+b.*sin(q2)-0.5.*L2.*(q11+q12).^2;
  .*sin(q2)+b.*cos(q2)+0.5.*L2.*(q21+q22);
  .*sin(q3)-0.5.*L2.*q13.^2;
  .*cos(q3)-0.5.*L2.*q25.^2;
  1.*L1.^2/12;
  2.*L2.^2/12;
  ;
  ;
  .*L5.^2/12;
  .5.*(q13+q14).^2.*L1;
  .5.*(q23-q24).*L1;
  .*cos(q7)-h.*sin(q7)-0.5.*L5.*(q13+q14+q17).^2;
  .*sin(q7)+h.*cos(q7)+0.5.*L5.*(q23+q24+q27);
  1.*q21-m1.*x1.*(b-g)-m1.*a.*y1;
  2.*q22+m2.*x2.*(d-g)-m1.*c.*y2;
  3.*q23+m3.*x3.*(f-g)-m3.*e.*y3;
  4.*q24+m4.*x4.*(k-g)-m4.*j.*y4;
  5.*q27+m5.*x5.*(v6-g)-m5.*v5.*y5;
  .*(b-g);
  .*(d-g);
  .*(f-g);
  .*(k-g);
  .*(v6-g);
  -s+r+s+v55)/(u+v+w+rr+v66);
```

Program A.14 : Program to calculate ZMP at quarter step

```
[...] L1,L2,L5,sx,sy,q1,q2,q3,q4,q7,q17,q27,m1,m2,m3,m4,m5,
...

...0.5*L1*cos(q1)+0.1*cos(pi/36);
...*L1.*sin(q1)+y+L2.*sin(pi-(q1+q2));
...0.5*L2*cos(pi-(q1+q2));
...0.5*L2*sin (pi-(q1+q2));
...*L2*cos(q3)+0.175*cos(pi/36);
...*L2*sin(q3)+0.175*sin(pi/36);
...0.5*L1*cos(pi-(q4+q3))+0.1*cos(pi/36);
...*sin(pi/36)-L2*sin(q3)-0.5*L1*sin(pi-(q4+q3));
...*cos(pi/36)+sx-0.5*L5*cos(q4+q3-q7);
...+sy-0.5*L5*sin(q4+q3-q7);
...1e06*t+3.6873*t.^2-4.3832*t.^3-1.726*t.^4;
...1e06+7.3746*t-13.1496*t.^2+6.904*t.^3;
...-.1975*t.^2+0.1456*t.^3-0.0555*t.^4;
...-.05*t+.4368*t.^2-.222*t.^3;
...4e08*t-4.6116*t.^2-5.2584*t.^3+1.999*t.^4;
...1e08-9.2232*t-15.7752*t.^2-7.996*t.^3;
...*t-.7014*t.^2-.8*t.^3-.304*t.^4;
...*t-1.4028*t+2.4*t.^2-1.216*t.^3;
...*q23.*cos(q4)+4.*q23+L1.*q24+L2.*(q13.^2).*sin(q4)-g.*cos(q3-q4);
...*q23.*sin(q4)-L2.*q13.^2.*cos(q4)-L1.*q13.^2-L1.*q14.^2-
...*q13.*q14-g.*sin(q3+q4);
...*cos(q1)+(h-g).*sin(q1)-0.5.*L1.*q11.^2;
...*sin(q1)+(h-g).*cos(q1)+0.5.*L1.*q22;
...*cos(q2)+b.*sin(q2)-0.5.*L2.*(q11+q12).^2;
...*sin(q2)+b.*cos(q2)+0.5.*L2.*(q21+q22);
...*sin(q3)-0.5.*L2.*q13.^2;
...*cos(q3)-0.5.*L2.*q23.^2;
...*L1.^2/12;
...*L1.^2/12;
...
...
...*L1.^2/12;
...+q13+q14).^2.*L1;
...+q23+q24).*L2;
...*cos(q7)-h.*sin(q7)-0.5.*L5.*(q13+q14-q17).^2;
...*sin(q7)+h.*cos(q7)+0.5.*L5.*(q23+q24-q27);
...q21+m1.*x1.*(b-g)-m1.*a.*y1;
...q22+m2.*x2.*(d-g)-m1.*c.*y2;
...q23+m3.*x3.*(f-g)-m3.*e.*y3;
...q24+m4.*x4.*(k-g)-m4.*j.*y4;
...q27+m5.*x5.*(v6-g)-m5.*v5.*y5;
...(b-g);
...(d-g);
...(f-g);
...(k-g);
...(v6-g);
...r+s-v551/(u+v+w+rr-v66);
```

APPENDIX – B : Joint angle trajectories

B.1 : For walking on 5 degree angle slope at full step length

Angle-theta1

$$\theta_1 = 0.8177 - 0.1203t^2 + 0.0678t^3 - 0.0155t^4 + 0.0012t^5$$

$$\theta_2 = -0.2406t + 0.2034t^2 - 0.062t^3 + 0.006t^4$$

$$\theta_3 = -0.2406 + 0.4068t - 0.186t^2 + 0.024t^3$$

Angle-theta2

$$\theta_1 = 1.4656 - 0.0563t^2 + 0.0317t^3 - 0.0073t^4 + 5.8137*10^{-4}t^5$$

$$\theta_2 = -0.1126t + 0.0951t^2 - 0.0292t^3 + 2.90685*10^{-3}t^4$$

$$\theta_3 = 0.125 - 0.2112t + 0.0972t^2 - 1.2906*10^{-2}t^3$$

Angle-theta5

$$\theta_1 = 1.2772 + 0.0625t^2 - 0.0352t^3 + 0.0081t^4 - 6.453*10^{-4}t^5$$

$$\theta_2 = 0.125t - 0.1056t^2 + 0.0324t^3 - 3.2265*10^{-3}t^4$$

$$\theta_3 = 0.125 - 0.2112t + 0.0972t^2 - 1.2906*10^{-2}t^3$$

Angle-theta6

$$\theta_1 = 1.4281 - 0.1927t^2 + 0.1086t^3 - 0.0249t^4 + 0.002t^5$$

$$\theta_2 = -0.3854t + 0.3258t^2 - 0.0996t^3 + 0.01t^4$$

$$\theta_3 = -0.3854 + 0.6516t - 0.2988t^2 + 0.04t^3$$

B.2 : For walking on 5 degree angle slope at half step length

Angle-theta1

$$\theta_1 = 0.8185 - 0.3233t^2 + 0.344t^3 - 0.1471t^4 + 0.0224t^5$$

$$\theta_2 = -0.6466t + 1.032t^2 - 0.5884t^3 + 0.112t^4$$

$$\theta_3 = -0.6466 + 2.064t - 1.7652t^2 + 0.448t^3$$

Angle-theta2

$$\theta_2 = 1.4639 + 4.6831*10^{-4}t^2 - 4.9824*10^{-4}t^3 + 2.1303*10^{-4}t^4 - 3.2396*10^{-5}t^5$$

$$\theta_2 = 9.3662*10^{-4}t - 1.4947*10^{-3}t^2 + 8.5212*10^{-4}t^3 - 1.6198*10^{-4}t^4$$

$$\theta_2 = 9.3662*10^{-4} - 2.9894*10^{-3}t + 2.5563*10^{-3}t^2 - 6.4792*10^{-4}t^3$$

Angle-theta5

$$\theta_5 = 1.4520 + 0.015t^2 - 0.0159t^3 + 0.0068t^4 - 0.001t^5$$

$$\theta_5 = 0.03t - 0.0477t^2 + 0.0272t^3 - 0.005t^4$$

$$\theta_5 = 0.03 - 0.0954t + 0.0816t^2 - 0.02t^3$$

Angle-theta6

$$\theta_6 = 1.0775 - 0.3433t^2 + 0.3652t^3 - 0.1562t^4 + 0.0237t^5$$

$$\theta_6 = -0.6866t + 1.0956t^2 - 0.6248t^3 + 0.1185t^4$$

$$\theta_6 = -0.6866 + 2.1912t - 1.8744t^2 + 0.474t^3$$

D.3 : For walking on 5 degree angle slope at quarter step length

Angle-theta1

$$\theta_1 = 0.8070 - 0.5648t^2 + 1.2291t^3 - 1.0958t^4 + 0.3452t^5$$

$$\theta_1 = -1.1296t + 3.6873t^2 - 4.3832t^3 + 1.726t^4$$

$$\theta_1 = -1.1296 + 7.3746t - 13.1496t^2 + 6.904t^3$$

Angle-theta2

$$\theta_2 = 1.4822 + 0.02t^2 - 0.0425t^3 + 0.0364t^4 - 0.0111t^5$$

$$\theta_2 = 0.04t - 0.1275t^2 + 0.1456t^3 - 0.0555t^4$$

$$\theta_2 = 0.04 - 0.255t + 0.4368t^2 - 0.222t^3$$

Angle-theta5

$\theta = 1.4520 + 0.1099t^2 - 0.2338t^3 + 0.2t^4 - 0.0608t^5$

$\theta = 0.2198t - 0.7014t^2 + 0.8t^3 - 0.304t^4$

$\theta = 0.2198 - 1.4028t + 2.4t^2 - 1.216t^3$

Angle-theta6

$\theta = 1.0775 - 0.7224t^2 + 1.5372t^3 - 1.3146t^4 + 0.3998t^5$

$\theta = -1.4448t + 4.6116t^2 - 5.2584t^3 + 1.999t^4$

$\theta = -1.4448 + 9.2232t - 15.7752t^2 + 7.996t^3$

D.4 : For walking on 10 degree slope angle at full step length

Angle-theta1

$\theta = 0.8112 - 0.1494t^2 + 0.0795t^3 - 0.0172t^4 + 0.0013t^5$

$\theta = -0.2988t + 0.2385t^2 - 0.0688t^3 + 6.5*10^{-3}t^4$

$\theta = -0.2988 + 0.477t - 0.2064t^2 + 2.6*10^{-2}t^3$

Angle-theta2

$\theta = 1.4709 - 0.0153t^2 + 0.0081t^3 - 0.0018t^4 + 1.3613*10^{-4}t^5$

$\theta = -0.0306t + 0.0243t^2 - 7.2*10^{-3}t^3 + 6.8065*10^{-4}t^4$

$\theta = -0.0306 + 0.0486t - 2.16*10^{-2}t^2 + 2.7226*10^{-3}t^3$

Angle-theta5

$\theta = 1.4520 - 0.0476t^2 + 0.0253t^3 - 0.0055t^4 + 4.2323*10^{-4}t^5$

$\theta = -0.0952t + 0.0759t^2 - 0.022t^3 + 2.1161*10^{-3}t^4$

$\theta = -0.0952 + 0.1518t - 0.066t^2 + 8.4644*10^{-3}t^3$

Angle-theta6

$$= 1.3578 - 0.1527t^2 + 0.0813t^3 - 0.0176t^4 + 0.0014t^5$$

$$= -0.3054t + 0.2439t^2 - 0.0704t^3 + 7*10^{-3}t^4$$

$$= -0.3054 + 0.4878t - 0.2112t^2 + 2.8*10^{-2}t^3$$

D.5 : For walking on 15 degree slope angle at full step length

Angle-theta1

$$= 0.8088 - 0.165t^2 + 0.0891t^3 - 0.0197t^4 + 0.0015t^5$$

$$= -0.33t + 0.2673t^2 - 0.0788t^3 + 7.5*10^{-3}t^4$$

$$= -0.33 + 0.5346t - 0.2364t^2 + 3*10^{-2}t^3$$

Angle-theta2

$$= 1.4681 + 0.0039t^2 - 0.0021t^3 + 4.7204*10^{-4}t^4 - 3.7036*10^{-5}t^5$$

$$= 7.8*10^{-3}t - 6.3*10^{-3}t^2 + 1.8881*10^{-3}t^3 - 1.8518*10^{-4}t^4$$

$$= 7.8*10^{-3} - 1.26*10^{-2}t + 5.6643*10^{-3}t^2 - 7.4072*10^{-4}t^3$$

Angle-theta5

$$= 1.4478 - 0.0424t^2 + 0.0229t^3 - 0.0051t^4 + 3.9790*10^{-4}t^5$$

$$= -0.0848t + 0.0687t^2 - 0.0204t^3 + 1.9895*10^{-3}t^4$$

$$= -0.0848 + 0.1374t - 0.0612t^2 + 7.958*10^{-3}t^3$$

Angle-theta6

$$= 1.0744 - 0.1381t^2 + 0.0746t^3 - 0.0165t^4 + 0.0013t^5$$

$$= -0.2762t + 0.2238t^2 - 0.066t^3 + 6.5*10^{-3}t^4$$

$$= -0.2762 + 0.4476t - 0.198t^2 + 2.6*10^{-2}t^3$$

APPENDIX – C: Simulation Programs

The following programs have been created with RoboWorks simulation software. The programs only display the main components used to build up the program in print view although each component is associated with relevant numerical figures.

Program E.1 : Simulation program of swing leg

- Root
  - slope
    - TRANSFORMATION START1
    - ROTATION1
    - TRANSLATION1
    - CUBE
    - TRANSFORMATION STOP1
  - Left Leg
    - TRANSFORMATION START13
    - TRANSLATION11
    - TRANSLATION14
    - ROTATION17
    - TRANSLATION25
    - CYLINDER
    - TRANSLATION48
    - ROTATION36
    - TRANSLATION67
    - CYLINDER
    - TRANSLATION100
    - ROTATION59
    - CUBE
    - TRANSFORMATION STOP39
  - Right Leg
    - CYLINDER
    - TRANSFORMATION START63
    - TRANSLATION221
    - ROTATION136
    - TRANSLATION228
    - CYLINDER
    - TRANSLATION239
    - ROTATION151
    - TRANSLATION254
    - CYLINDER
    - TRANSLATION273
    - ROTATION181
    - CUBE
    - TRANSFORMATION STOP61

Program E.2 : Simulation program of stance leg

- Root
  - GROUP1
    - TRANSFORMATION START1
    - TRANSLATION1
    - ROTATION1
    - CUBE
    - TRANSFORMATION STOP1
  - GROUP2
    - TRANSFORMATION START11
    - CYLINDER
    - TRANSLATION13
    - CUBE
    - ROTATION13
    - TRANSLATION20
    - CYLINDER
    - TRANSLATION31
    - ROTATION28
    - TRANSLATION46
    - [redacted]

Program E.3 : Simulation program for simulation of both legs

- Root
  - Slope
    - TRANSFORMATION START9
    - ROTATION92
    - TRANSLATION146
    - CUBE
    - TRANSFORMATION STOP1
  - Both Legs
    - TRANSFORMATION START12
    - CYLINDER
    - TRANSLATION55
    - CUBE
    - ROTATION41
    - TRANSLATION38
    - CYLINDER
    - TRANSLATION35
    - ROTATION35
    - TRANSLATION44
    - CYLINDER
    - TRANSLATION63
    - TRANSLATION91
    - CYLINDER
    - TRANSLATION114
    - ROTATION71
    - TRANSLATION141
    - CYLINDER
    - TRANSLATION170
    - ROTATION92
    - TRANSLATION199
    - CYLINDER
    - TRANSLATION260
    - ROTATION109
    - CUBE
    - [redacted]

Program E.4 : Simulation program of bipedal robot lower body with torso

- Root
  - Slope
    - TRANSFORMATION START9
    - ROTATION92
    - TRANSLATION146
    - CUBE
    - TRANSFORMATION STOP1
  - Both Legs
    - TRANSFORMATION START12
    - CYLINDER
    - TRANSLATION55
    - CUBE
    - ROTATION41
    - TRANSLATION38
    - CYLINDER
    - TRANSLATION35
    - ROTATION35
    - TRANSLATION44
    - CYLINDER
    - TRANSLATION63
    - TRANSLATION91
    - CYLINDER
    - TRANSFORMATION START61
    - TRANSLATION114
    - ROTATION161
    - TRANSLATION325
    - CYLINDER
    - TRANSFORMATION STOP49
    - TRANSFORMATION START81
    - TRANSLATION300
    - CYLINDER
    - TRANSLATION291
    - ROTATION127
    - TRANSLATION286
    - CYLINDER
    - TRANSLATION287
    - ROTATION124
    - TRANSLATION300
    - CYLINDER
    - TRANSLATION323
    - TRANSLATION406
    - ROTATION143
    - CUBE
    - TRANSFORMATION STOP61