MATHEMATICAL MODELING AND NEURAL NETWORK CONTROLLER DESIGN FOR BIPED ROBOT'S HUMAN LIKE WALK

Ph.D. THESIS

by

RUCHI PANWAR

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247667 (INDIA)
JULY, 2019

MATHEMATICAL MODELING AND NEURAL NETWORK CONTROLLER DESIGN FOR BIPED ROBOT'S HUMAN LIKE WALK

A THESIS

Submitted in partial fulfilment of the requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

by

RUCHI PANWAR

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE–247667 (INDIA)
JULY, 2019

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in this thesis entitled, "**MATHEMATICAL MODELING AND NEURAL NETWORK CONTROLLER DESIGN FOR BIPED ROBOT'S HUMAN LIKE WALK**" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Mathematics of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during a period from July, 2013 to July, 2019 under the supervision of Dr. N. Sukavanam, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institution.

(RUCHI PANWAR)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(N. Sukavanam)
Supervisor

The Ph.D. Viva-Voce Examination of **Mrs. Ruchi Panwar,** Research Scholar, has been held on .....

Chairman, SRC **Signature of External Examiner**

This is to certify that the student has made all the corrections in the thesis.

Signature of Supervisor **Head of the Department**

Dated:.....

Abstract

This thesis is concern with gait generation methods for stable walk of flat footed and toe footed biped robot models. Methods are proposed to generate suitable trajectory, which can easily adapt the changes in the boundary conditions/constraints during walk. Different type of approaches such as polynomials, Feedforward Neural Network(FNN) and Wavelet Neural Network(WNN) are considered for some biped models walking on uneven surfaces and for avoiding obstacles. To avoid stable leg's knee bending and for more stable walk, lateral upper body motion is considered. Force/torque control for walk is designed by developing the dynamic equations of the biped model.

The thesis is divided into 7 chapters which are briefly described below:

Chapter 1 is introductory with a brief literature review in the area of robotics and neural networks. We have discussed some biped models and also the inherent challenges associated with the robots' stable walk. Finally, a summary of the thesis is presented.

Chapter 2 gives some basics and preliminaries which are used in subsequent chapters.

Chapter 3 focuses on stable FNN trajectory tracking of flat footed biped robot with upper body motion. Trajectories using cubic spline are generated for ankle joints, hip joints and upper body so that the resulting walk is stable. Here, the effects of different lateral upper body motions of the flat footed robot on ZMP stability is analyzed for plane surface walking. The inverse kinematics of the ankle and hip trajectories are solved using FNN. Further, simulations are done using Matlab2010b.

Chapter 4 proposes polynomial based trajectory generation algorithm of a robot model for stable human like gait considering the upper body motion, movable foot and active toe. This approach allows the smooth transition between walking phases namely, single support phase and double support phase. ZMP stability is analyzed for plane and uneven surface walking of the toe footed model by taking into account the lateral upper-body movements

along with the planned motion trajectories.

Chapter 5 proposes and compares FNN and WNN based approaches for smooth trajectory generation under given constraints. The trajectory generation procedure is derived from semi-supervised NNs for given boundary conditions without assigning any path in advance. The trajectories generated by using proposed approaches can be modified according to the constraints value at any instant of time during tracking. Further, these approaches are used for the gait generation of a 5 DOF flat footed biped to walk on flat terrain in 3-dimensional space. The suitability of the proposed approaches is studied using ZMP stability criteria and simulations have been carried out using Matlab2014a.

In Chapter 6, a biped robot model with flat foot is considered. The dynamic equation of this model is derived and a PD controller for stable walk is presented. FNN approach proposed in Chapter 5 is used for smooth and dynamically stable trajectory generation and the results are compared with polynomial approach. Simulation results (using Matlab2014a) show that this model can cross over obstacles of different heights and cross over a ditch by adjusting the step height and step length in ankle trajectory at any instant during tracking.

Conclusions, limitations and future work have been outlined in Chapter 7.

Acknowledgement

I pay immense gratitude to the Almighty God for blessing me with the fearlessness to face the complexities of life and complete this thesis effectively. I am fortunate enough to have some extraordinary people in my life who have rendered invaluable love and support during every challenging phase.

At this moment of achievement, I am highly grateful to my supervisor Prof N. Sukavanam, Head of the Department, Department of Mathematics, I.I.T. Roorkee, whose timely guidance, discriminating remarks and suggestions at diverse stages helped me sharpen the ideas throughout my research work. This feat was possible only because of the unconditional support provided by Sir during tough times in the Ph.D. pursuit. He has always made himself available to clarify my doubts despite his busy schedules and I consider it as once in a lifetime opportunity to do my doctoral programme under his valuable guidance and learn from his research expertise. He has been an excellent example of a mathematician, professor and moreover a person with high moral values. By being in his shelter has not only helped me to grow as a researcher but also as an individual.

I express my sincere respects to Prof. R. C. Mittal (Ex Head), Prof. V. K. Katiyar (Ex Head), Prof N. Sukavanam, Head, Department of Mathematics, IIT Roorkee for providing valuable advice and other infrastructure facilities during my research work. I am thankful to my research committee members for their guidance, cooperation and numerous important remarks that have helped me in enhancing the quality of my research work. I consider it as an honor to be a part of Department of Mathematics, IIT Roorkee. I express my earnest regards to all the faculty and non teaching staff members of Department of Mathematics for their unstinted support at all levels. I gratefully acknowledge the MHRD for providing financial assistance for my work at IIT Roorkee.

I have no words to define my sincere thanks to my family members, Buaji and relatives

for their blessings, patience and moral support. In spirit, this thesis belongs more to them then it does to me. I owe it to my maternal and paternal grand parents, my sisters and in law sisters specially Geeta, Chesta, Arzoo, Dipika and Preeti for solving all my worries by taking care of my beloved baby Vanya(Pihu). Thanks to my whole batch and juniors specially Alka, Neha, Nutan, Lavina, Karuna and Sumit for holding my hand through all the PhD madness, for supporting me and encouraging me in every problem. Warm thanks to my partner Sunil for inspiration, cooperation, support and patience in all dimensions and being a true friend in every situation.

Thank you everyone. I dedicate this work to you all.

Ruchi Panwar
IIT ROORKEE

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Robots are increasingly used in industry, agriculture, space, military and medical applications. The study of bipedal walk will help in the development of more sophisticated humanoid robots. Researchers give more importance to humanoid robots because of their resemblance to human structure. They are able to climb up and down on stairs, walk on narrow places, jump and perform most of the work better than humans. Stable biped locomotion is a challenging problem in robotics. A typical problem is the instability produced by violent transitions between walking phases, particularly when a swing leg impacts the ground. During walk, falls commonly occur when the walking speed increases or the terrain conditions change. However, considerable work has been done on locomotion in literature but more work needs to be done. As robots will be indispensable in our daily life, it is expected that the human-robot interaction is and will be an area of active research.

1.1 Introduction to Bipedal Robots

Initially, the study on artificial hands and arms began in 1967 for the development of robots that can perform work intelligently as a manual skilled labor [4, 27, 79, 85]. Since 70s, many studies were done on biped robot. Vukobratovic et al.(1972) [96] was one of the pioneering researchers of biped locomotion and Zero Moment Point(ZMP) concept. The inverted pendulum model was developed by Miura et al.(1980) [64] on bipedal gait and it could achieve the dynamically stable walk. Zheng et al.(1990) [103] developed gait synthesis for the SD-2 biped robot to climb sloping surface.

The ZMP concept is applied by Li(1991) [50] to control the motion of 12 degrees of freedom (DOF) WL-12RIII biped robot. Kajita et al.(1992) [37] proposed an potential energy conserving orbit concept to formulate new control laws on the biped. Goswami

et al.(1999) [24] introduced a point called the foot rotation indicator (FRI) point on the contact surface which is the net ground reaction to keep the foot stationary.

Lim et al.(2000) [55] modeled and studied a balance control of 43 DOF biped robot (WABIAN-RII) for human like walk. Firstly, the smooth motion is designed of its lower limbs using a set of walking parameters, and the motion of its trunk and waist is derived using an iterative method. Finally, a controller is applied to the planned walk. Gienger(2001) [23] proposed the design and the controller for a 3 dimensional dynamically stable walk of the bipedal robot. Plestan et al.(2003) [76] demonstrated asymptotically stable walk of a planar 5 link underactuated bipedal robot model. To verify disturbance rejection capability, different perturbations were introduced in walk patterns. Mousavi et al.(2007) [65] focused on simulation and control of the biped robot model on horizontal, ascending and descending surfaces. The robot path using the given break points is interpolated using mathematical simulation.

1.1.1 Types of Bipedal Robots

The lower limb model named WL-1 was introduced in 1966-1967 [8, 9, 54]. It was an artificial lower limb which was made on the basis of locomotion of the lower limb's analysis. Further, a master type walking machine WL-3 was developed in 1968 to 1969 which was a mechanical model for lower limbs. It produced human-like movement for the single and double support phases and it could also sit and stand up. The WAP-3 was developed in 1971 which was a light weighted model for bipedal walking. It was able to walk on a slope or staircase, and it could also turn while walking. The static walking was realized by a heavy model WL-5. This was developed in 1970-1972 and controlled by a mini computer. In WABOT-1 (45sec/step), WL-5 was used as lower limbs. In 1980-1981 the computer aided design system was developed for artificial limbs.

All these models work in 2 dimensional space and walk slowly using static stability with large feet. In the 1980s, the bipedal robot BIPER [63] was developed to perform dynamic walk with larger feet to maintain the balance. Dynamic walk was realized in 1984 with a model WL-10RD (1.3 sec/step). In 1991, the MELTRAN-II was constructed by Kajita et al. [36] using the inverted pendulum concept. A series of bipedal robots had been started by Honda since the 90s like 25 DOF ASIMO robot [80] with 1.4 meter height. The

Toyota biped robot can play musical instrument and the first kid size biped robot SONY Qrio [33] was built in 2003. Recently, the HRP series [39] robots is built by AIST institute and the Kawada Industries. The HRP-2 robot has 30 DOF with 1.52m height and 58 kg weight. The KHR-1 has 21 DOF with no hands and head [73] and KHR-2 has 41 DOF and also have human like walk. In 2010, HUBO is refined with its human like movements and features. The Nao robot [25] is built by the company Aldebaran which is a 58-cm tall humanoid robot and it is successfully used in computer and science classes for education. Humanoid robots Johnnie/LOLA [23] by University of Munich Germany is powered by linear actuators. LUCY from Vrije Universiteit Brussel uses pneumatic muscle, and iCub [69] by University of Genova Italy is a small biped robot. The DARPA [35] Robotics Competition 2014 is the center of attraction for the development of autonomous bipeds for exploring disaster field and used in search and rescue missions. Three years later, in 2016, italian institute of technology (IIT) introduced a new high-performance humanoid for the realistic environment: WALKMAN [93]. Atlas [57] introduced by Boston Dynamics is able to balance when jostled or pushed and can able to walk on different surfaces. Atlas's ability to balance while performing tasks and greatly expanding its reach and workspace allows it to work in a large volume with small feet.

1.2 Challenges Associated with Human Walk

How human gait works is still an active research field. Many researchers have developed different types of biped robots which can walk with varying speed[31-36] but still a great amount of theoretical work needs to be done in this field. To realize human like walk of a biped robot, trajectory planning is considered the most important factor. For a biped robot to walk, good balance and stability is also necessary. Some biped robot model have flat feet, and their knees and upper body are bent during walking which look like a Neanderthal walking. They bend their knees and upper body, all the time during double support phase of locomotion to balance and to shift the body weight from swing leg to the stable leg [15, 21, 31, 38, 67, 80]. This walk not only looks unnatural but also leads to high torque on knee, hip and ankle joints [59, 88, 100], whereas humans walk in a different manner. They do not shorten the stable leg, but over-extend their swing leg with the help of foot rotation [48, 68, 99].

To achieve human gait, it is important to control the motion of the robot and to plan suitable trajectories for the robot joints during the walk. For a smooth trajectory, path velocity and acceleration should remain smooth from the start to the final point. Smooth trajectory is needed for accuracy, efficiency in repetitive tasks and high productivity for industrial work and also it is also beneficial for biped to avoid jerks. Accurate kinematic and dynamic models are required for efficient locomotion.

Firstly a brief literature survey related to this research is presented.

1.3 Literature Survey

1.3.1 Various Path Planning Techniques

Trajectory generation problems have been extensively studied in the literature [1, 6, 10, 15, 22, 31, 66, 67, 78, 86, 87, 94, 100]. There are several numerical interpolation and convolutional methods for smooth motion to generate a collision-free trajectory from an initial to a final position [6, 10, 28, 40, 58, 75, 78, 87].

Researchers have designed various trajectories for the hip and the foot joints for a stable walk. Many techniques have been adopted for walk planning of biped robot [22, 31, 66, 67, 100]. The authors in [15, 31, 94, 100] presented a cubic polynomial interpolation algorithm to implement the biped walk. Also, higher degree polynomials have been used for the smooth trajectory generation to maintain the continuity of the velocity and acceleration [87].

Narvez-Aroche et al. [67] have obtained a kinematic model for 12 DOF biped robot which gives satisfactory results for the position, velocity and acceleration control. They also presented the steps for the kinematic modeling and ZMP computation method.

Xiaoguang and Ruyi [100] presented a gait plan for NCEPU-I humanoid robot model with the bar linkage. The hip trajectory and the swing leg's ankle trajectory are planned for slope gait. This method can be applied to different angles of high efficiency.

Cuevas et al. [15] used a polynomial trajectory generation algorithm (PTA) on a 10 DOF biped robot from initial to final conditions for smooth transition during walking phases. The joint trajectories are generated for various surface conditions.

Some robots are also able to cross the obstacles of different heights [15, 51, 72, 90, 104].

Stasse et al. [90] presented a HRP-2 humanoid robot that can dynamically step over a 15 cm obstacle within 4 secs using quasi static stability with 3 cm stability margin.

On the other hand, Li et al. [51] planned a gait by motion capture system towards overcoming the obstacle during walk of the humanoid robot. Force sensors are used to calculate the ground reaction forces.

However, these methods don't work well in more complex domain. So, the development of bipedal robot, which can walk on uneven terrain and can adjust its gait/step according to uncertain environment during tracking is one of the challenging fields of research. Hence, many researchers have investigated different soft computing techniques in different areas recently.

1.3.2 Soft Computing Techniques

In recent time due to the complexity and computational challenges of problems, soft computing techniques such as Neural Network (NN), Genetic Algorithm (GA), Fuzzy Logic (FL) and Particle Swarm Optimization (PSO) etc are popularly considered by several authors [2, 3, 7, 41, 43, 52, 55, 59–61], to solve various problems in different areas of research. These techniques are effective and have learning capability to solve some complex, uncertain and real world problems. Biped robot's gait generation is difficult due to its complexity in model development and imprecision in data collection and uncertainty in walking surfaces. To overcome these challenges, gait can be best modeled using the soft computing techniques.

Capi et al. [12] used splines for angle trajectory generation based on GA, for stable walk and going up-stair with the minimum energy consumption. A radial basis function neural network is considered for the real time application. Simulations are carried out on Bonten-Maru I biped robot.

Vundavilli et al. [97] used two approaches namely, GA-NN and GA-FLC to generate stable gait of biped for the staircase. GA, is used offline to optimize the NN's weights and to optimize the knowledge bases of FLC.

Nada Kherici et al. [43] used particle swarm optimization (PSO) to generate articulation angles for stable gait by using the center of mass (COM) constraint of the 10 DoF biped robot.

Sarkar and dutta [82] have used GA for energy efficient gait of compliant links bipedal robot in case of various step lengths and slopes. Gait parameters are varied in a range, and using these parameters different joint trajectories are generated. GA was used to find the minimum energy consumption walking trajectory with ZMP stability.

1.3.3 Neural Network as Universal Approximator

Recently, NN has generated a significant interest among soft computing techniques for its learning ability in the control field, natural language processing, speech and image recognition. NN's capability and flexibility of learning by example, generalizing property, smaller information prerequisite, quick real time operation and simplicity of implementation characteristics have made them popular in every field of research.

NN has been used (i) to find numerical solution of differential equations [49, 71, 91], (ii) to approximate multi dimensional non-linear systems [19, 41, 44], (iii) to map the input-output representation of an unknown system [89] and (iv) to identify and control dynamical systems [14, 46, 47, 70].

Several NN models also have been proposed to resolve the trajectory planning problems [1, 29, 30, 56, 60, 62, 97, 98]. Liu et al. [56] used online trajectory generation method based on neural oscillators for adaptive walking control of biped robot. In [60], SVM controller is used to predict the truck trajectory using foot and hip trajectories as input to generate a balanced gait. In some papers, trajectories are generated by NN, using sensory feedback of data [56, 105], or by collecting human data [84].

NN approach is effective in generating a trajectory, since it can give instant solution at any desired number of points in less computational time and memory, once the networks have been trained.

NN can also be used to find inverse kinematic solutions for biped robots. In [5, 16, 17, 32], some random samples of joint angles are taken from specified range and then used in forward kinematic equations to calculate the end effector's position and orientation for each set of joint angles. These pairs of joint angles and position/orientation are called training data. NN function is trained using supervised learning for these training data (approximately 1000) by giving position and orientation as inputs and the corresponding joint angles as desired outputs. Further, NN (at least 100 neurons in hidden layer) gives inverse

kinematics solutions for any position in workspace. De et al. [16] found the inverse kinematic solutions for humanoid robot using supervised learning approach for training data. Two NNs are used, one for DSP and another for SSP, to learn the most stable positions from the range of all the possible solutions on basis of COM and no hip roll rotation.

1.3.4 Stable Gait/Locomotion on Generated Trajectory

The objective of the present day research is, not only to design suitable trajectories for gait generation on complex surfaces but also to make sure that the generated gait is dynamically balanced. Erbatur et al. [18] used the cart-table model differential equation for COM trajectory. Vukobratovic et al. [95] used the ZMP concept for stable walk of the biped robots. The most widely used [15, 18, 31, 51, 53, 59, 67, 68, 74, 88, 90, 94, 100, 101, 104] dynamic balance criterion is the ZMP. Some researchers have assumed a pre-determined ZMP trajectory [68, 104]. It can be either be a fixed ZMP [31, 67, 80] typically at the center of the sole in the single support phase, or a moving ZMP which moves from one foot to another periodically in the supported region during locomotion [18, 30, 56, 59, 82, 101, 104].

Park [74] has proposed a fuzzy logic ZMP trajectory to reduce the swing motion of the trunk for the stable walk of biped robot. The trunk trajectory of 7 DOF biped robot is calculated by solving a differential equation with known ZMP and leg trajectories in sagittal plane.

Erbatur and Kurt [18] presented an approach for the iterative trajectory generation using a Linear Inverted Pendulum Model (LIPM) algorithm by specifying a desired ZMP reference trajectory. These techniques are tested and simulated on a 12 DOF biped robot model. Narvaez et al. [67] presented a kinematic analysis of a 12 DOF biped robot with ZMP stability.

In Zhu et al. [104], the optimal allowable ZMP variation region is used to generate different locomotion with respect to different surface conditions by modifying the ZMP in lateral direction. Further, gait parameters are optimized for maximum stability margin.

1.3.5 Effect of Upper Body and Toe Joint on ZMP Stability

To obtain the desired ZMP reference trajectory, the suitable joint motion are required. The knee bending problem can be alleviated by allowing a vertical body motion (VBM) [59, 88]. Takanishi et al. [92] presented a method for generating upper body motion by transforming the ZMP reference trajectories into the Fourier series. Lim et al. [53] described a walking pattern of 16 DOF WABIAN-2LL biped robot with stretched knee which could moves around an object using a hip bending motion. Shin et al. [88] proposed a locomotion planning of a biped model with vertical body movement to relieve from knee bending problem with allowable ZMP region (AZR). Here, a control system is established with 3 Mass Inverted Pendulum Mode (3MIPM). Gait parameters are optimized for minimum energy consumption with various ZMP in AZR. In Liu et al. [59] a knee stretched walking pattern is proposed involving three dimensional motion of upper body based on ZMP dynamic balance criterion. The proposed walking pattern is modified according to sensory feedback to stabilize the Nao robot. Vertical upper body point mass motion is effective in attaining a wide range of step lengths with improved push recovery behavior of biped robot [22, 102].

In human walk, first the heel of the swing leg contacts the ground, then the sole and finally the toe contact the ground [18, 99]. For humanoid H6, Nishiwaki et al. [68] added the extension of foot with toe joints attachment to avoid from knee bending, to speed up the walking and to enable the robot to take higher steps.

Furthermore, dynamic control is essential part in robotics. Torque/forces are required for the implementation of the proposed walking patterns. Different dynamic model with controller for stable locomotion are presented in [7, 11, 41, 45, 81, 98]. Vundavalli et al. [97] derived dynamic equations and optimal control for 7 DOFs biped robot. Sarkar and dutta [82] presented energy efficient controlled gait with compliant links for a 12 DOF bipedal robot model.

This thesis is concern with gait generation methods for stable walk of flat footed and toe footed biped robot models. Methods are proposed to generate suitable trajectory, which can easily adapt the changes in the boundary conditions/constraints during walk. Different types of approaches such as polynomials, Feedforward Neural Network(FNN) and Wavelet Neural Network(WNN) are considered for some biped models walking on uneven surfaces

and for avoiding obstacles. To avoid stable leg's knee bending and for more stable walk, lateral upper body motion is considered. Force/torque control for walk is designed by developing the dynamic equations of the biped model.
The thesis is divided into 7 chapters which are briefly described below:

Chapter 1 is introductory with a brief literature review in the areas of robotics and neural networks. We have discussed some biped models and also the inherent challenges associated with the robots' stable walk. Finally, a summary of the thesis is presented.

Chapter 2 gives some basics and preliminaries which are used in subsequent chapters.

Chapter 3 focuses on stable FNN trajectory tracking of flat footed biped robot with upper body motion. Trajectories using cubic spline are generated for ankle joints, hip joints and upper body so that the resulting walk is stable. Here, the effects of different lateral upper body motions of the flat footed robot on ZMP stability is analyzed for plane surface walking. The inverse kinematics of the ankle and hip trajectories are solved using FNN. Further, simulations are done using Matlab2010b.

Chapter 4 proposes polynomial based trajectory generation algorithm of a robot model for stable human like gait considering the upper body motion, movable foot and active toe. This approach allows the smooth transition between walking phases namely, single support phase and double support phase. ZMP stability is analyzed for plane and uneven surface walking of the toe footed model by taking into account the lateral upper-body movements along with the planned motion trajectories.

Chapter 5 proposes and compares FNN and WNN based approaches for smooth trajectory generation under given constraints. The trajectory generation procedure is derived from semi-supervised neural network for given boundary conditions without assigning any path in advance. The trajectories generated by using proposed approaches can be modified according to the constraints value at any instant of time during tracking. Further, these approaches are used for the gait generation of a 5 DOF flat footed biped to walk on flat terrain in 3-dimensional space. The suitability of the proposed approaches is studied using ZMP stability criteria and simulations have been carried out using Matlab2014a.

In Chapter 6, a biped robot model with flat foot is considered. The dynamic equation of this model is derived and a PD controller for stable walk is presented. FNN approach proposed in Chapter 5 is used for smooth and dynamically stable trajectory generation and the

results are compared with polynomial approach. Simulation results (using Matlab2014a) show that this model can cross over obstacles of different heights and cross over a ditch by adjusting the step height and step length in ankle trajectory at any instant during tracking.

Conclusions, limitations and future work have been outlined in Chapter 7.

Chapter 2: Preliminaries

2.1 Homogeneous Transformation Matrix

Homogeneous transformation is used to describe the position and orientation relationship between two coordinate frames. The kinematics of a robot is concerned with the relationship between co-ordinate frames attached to the object to be handled/environment and the joints of the robot. Let G be the universal frame and M be the moving frame, then the position and orientation of frame M with respect to frame is given by a 4×4 homogeneous transformation matrix ${}^{G}T_{M}$ as

$$ {}^{G}T_{M} = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{2.1} $$

where R is a rotation matrix representing the orientation and P is a translation vector with respect to the frame G.

The inverse of ${}^{G}T_{M}$ is given by

$$ [{}^{G}T_{M}]^{-1} = {}^{M}T_{G} = \begin{bmatrix} R' & -R'P \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.2} $$

where R' denotes the transpose of R.

If M is obtained from G only by translation along a vector $\vec{P} = (a, b, c)$ then

$$ {}^{G}T_{M} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3} $$

which can be represented as $Ta(\vec{P})$. If M is obtained by rotating G according to a rotation matrix R, then

$$^{G}T_{M} = \begin{bmatrix} R_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2.4}$$

The homogeneous transformation matrices representing rotation by an angle θ about the x, y, z axes respectively, are given by

$$R(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R(y,\theta) = \begin{bmatrix} C\theta & 0 & S\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta & 0 & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R(z,\theta) = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2 Forward and Inverse Kinematics

Kinematics in robot mechanism describes the motion of a body without considering the torque/force acting on it during the motion. For n link manipulator, a relation between the joint variables and the position/orientation of the end effector with respect to the base is the kinematic equations. The robot kinematics problem is classified into forward and inverse kinematics.

2.2.1 Kinematics of 2 link Manipulator

Consider the 2-link manipulator OAB, where O is at the origin of the XY coordinate frame. Let (x, y) denotes the coordinate of the end-effector B and let (θ_1, θ_2) be the joint angles as shown in the Figure 2.1. From Figure 2.1, it can be easily seen that

$$x = l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) \tag{2.5}$$

$$y = l_1 sin\theta_1 + l_2 sin(\theta_1 + \theta_2) \tag{2.6}$$

where l_1 and l_2 are length of the links.

The above set of 2 equations is called kinematic equations of the given manipulator.

Finding the end effector position (x,y) for given joint angles (θ_1, θ_2) is called forward kinematics while finding (θ_1, θ_2) for given (x,y) is called inverse kinematics.

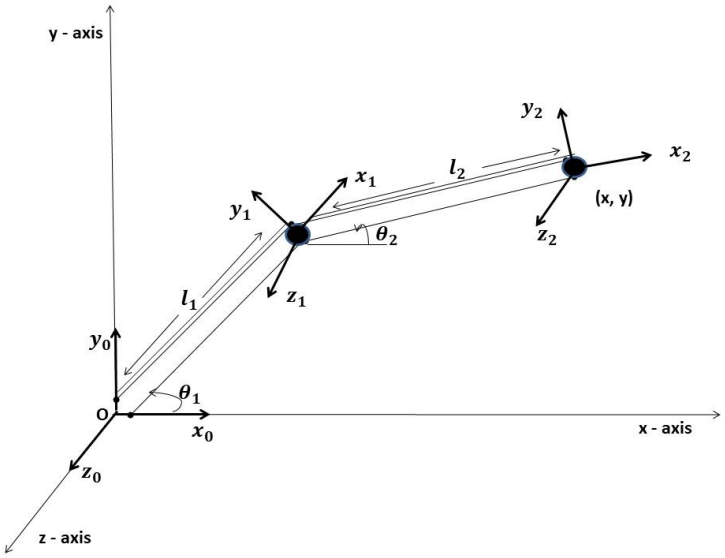


Figure 2.1: 2 link manipulator

The inverse kinematic solutions can be easily obtained from equations (1) and (2) as follows:

$$\theta_2 = \cos^{-1}\left[\frac{x^2 + y^2 - {l_1}^2 - {l_2}^2}{2l_1 l_2}\right] \tag{2.7}$$

$$\theta_1 = \sin^{-1}\left[\frac{y(l_1 + l_2\cos\theta_2) - x l_2 \sin\theta_2}{x^2 + y^2}\right] \tag{2.8}$$

As number of the links is increased, kinematics for a robot becomes difficult to determine. Denavit-Hartenberg proposed a matrix method using translational and rotational relationships between adjacent links for the coordinate of an end effector with respect to the base frame.

2.2.2 Denavit-Hartenberg (D-H) Procedure

A robot manipulator is a sequence of links connected by either revolute or prismatic joints. Denavit Hartenberg proposed an approach by utilizing homogeneous transformation matrix to describe the spatial geometry of the links with respect to the base reference frame. Firstly, the coordinate frames are assigned on each joint and then DH parameters ($\theta_i, d_i, a_i, \alpha_i$) for each joint i are calculated as in [20, 79, 83] where θ_i is the joint angle, d_i is the joint distance, a_i is the link length and α_i is the link twist angle for each $i = 1, 2, ..., n$.

The position and orientation coordinates of the ith frame can be expressed in the (i-1)th

frame by a 4×4 homogeneous transformation matrix after substituting the corresponding DH parameters. The matrix ${}^{i-1}T_i$ denotes the position and orientation of the ith frame with respect to (i-1)th frame which is

$$\begin{aligned} {}^{i-1}T_i &= R(z_{i-1}, \theta_i)Ta(0,0,d_i)Ta(a_i,0,0)R(x_i,\alpha_i) \\ &= \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\alpha_i C\theta_i \\ S\theta_i & C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3\times 3} & P_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix} \end{aligned}$$

The Arm Matrix for n-link manipulator is

$${}^0T_n = {}^0\ T_1(q_1){}^1T_2(q_2){}^2T_3(q_3)\ldots{}^{(n-1)}T_n(q_n) = \prod_{i=1}^{n} {}^{i-1}T_i$$

which represent the position and orientation of end effector with respect to base frame. The inverse of ${}^0T_n$ is given by

$${}^nT_0 = {}^0T_n{}^{-1} = {}^n\ T_{(n-1)}(q_n){}^{(n-1)}T_{(n-2)}(q_{(n-1)})\ldots{}^2T_1(q_2){}^1T_0(q_1)$$

2.2.3 Inverse Kinematics

The main problem for the control of motion in robotics, is to find accurate and reliable inverse kinematic solutions. Solving the inverse kinematic problems for robots is quite challenging task because of geometry complexity, multiple solutions, singularities and nonlinear trigonometric equations occurring between Cartesian space and joint space. Traditional methods such as geometric, iterative and algebraic are inadequate if the joint structure of the manipulator is more complex. Disadvantages of analytical approaches to inverse kinematic solutions are that they do not adapt well to the changes in the system parameters. In order to overcome disadvantages of these techniques, Neural Networks(NN) and Optimization methods have been widely used for inverse kinematics problems in robotics.

2.3 Feedforward Neural Network(FNN)

A two layer FNN with M input units, m output units and N hidden layer neurons is shown in Figure 2.2. The relation between input vector $s \in \mathbb{R}^M$ and the output vector

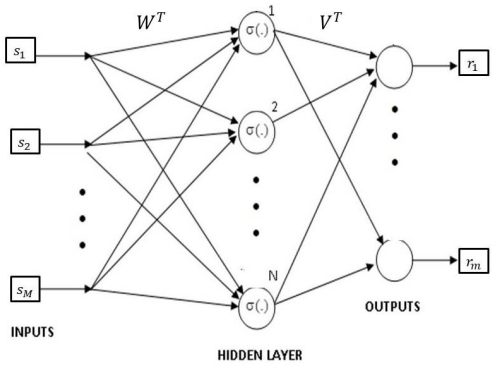


Figure 2.2: Feedforward neural network

$r = f(s) \in \mathbb{R}^m$ is given in the following equation:

$$r_i = f_i(s) = \sum_{j=1}^{N} \left[v_{ij} \sigma \left(\sum\nolimits_{k=1}^{M} w_{jk} s_k \right) \right]; \quad i = 1, 2, .., m \tag{2.9}$$

where $\sigma(.)$ is the activation function for the hidden-layer neurons, w_{jk} and v_{ij} are interconnection weights from input layer to hidden layer and hidden layer to output layer respectively.

Sigmoid, Signum, tangent hyperbolic and gaussian are some activation functions generally used in NN literature. Sigmoid activation function is given by

$$\sigma(s) = \frac{1}{1 + e^{-s}}. \tag{2.10}$$

Define the weight matrices as W=$[w_{jk}]_{\substack{j=1,2,...,N \\ k=1,2,...,M}}$ and V=$[v_{ij}]_{\substack{i=1,2,...,m \\ j=1,2,...,N}}$. Then equation(2.1) can be rewritten as

$$r = f(s) = V\sigma(Ws), \tag{2.11}$$

where $r = [r_1, r_2, ..., r_m]'$, $s = [s_1, s_2, ..., s_M]'$ and $\sigma(s) = [\sigma(s_1), ..., \sigma(s_M)]'$.

NN function approximation property: According to the NN function approximation property, for a given smooth function f on a compact set $\Omega \subset R^M$ and arbitrary $\varepsilon > 0$, there exists N hidden layer neurons and weight matrices W and V such that

$$r = f(s) = V\sigma(Ws) + \varepsilon_1(s), \tag{2.12}$$

where NN approximation error $\varepsilon_1(s)$ satisfies $||\varepsilon_1|| < \varepsilon$.

If ε is sufficiently small then, an estimation of $f(s)$ in terms of neural network can be written as

$$\hat{f}(s) = \hat{V}\sigma(\hat{W}s), \tag{2.13}$$

where $\hat{W}$ and $\hat{V}$ are the approximations of W and V respectively that are provided by some online weight tuning algorithm.

Error backpropagation Algorithm: This is a common weight tuning algorithm based on gradient descent algorithm. Let r_d be the desired NN output and r is the actual output. Then the backpropagated error is $E = r_d - r$. Now, the weights are updated to minimize the error E by the backpropagation algorithm given by:

$$v_{ij}^{(I+1)} = v_{ij}^{(I)} - \alpha_1 \frac{\partial E}{\partial v_{ij}^{(I)}} \tag{2.14}$$

$$w_{jk}^{(I+1)} = w_{jk}^{(I)} - \alpha_2 \frac{\partial E}{\partial w_{jk}^{(I)}} \tag{2.15}$$

where α_1 and α_2 are learning rates and I is number of iterations.

2.4 Wavelet Neural Network(WNN)

The structure of WNN is shown in Figure 2.3. The relation between input signal vector $x \in \mathbb{R}^M$ and the output vector $y = f(x) \in \mathbb{R}^m$ is given in the following equation:

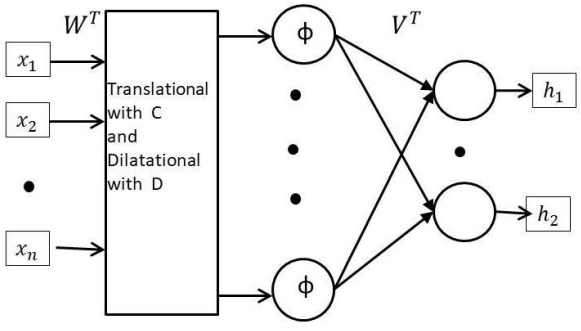


Figure 2.3: Wavelet neural network

$$y_i = h_i(x) = \sum_{j=1}^{N} \left[v_{ij}\phi\left(\sum_{k=1}^{M} \frac{(w_{jk}x_k - c_k)}{d_k} \right) \right]; \quad i = 1, 2, ..., m \tag{2.16}$$

where $\phi(.)$ is the activation function for the hidden-layer neurons, w_{jk} and v_{ij} are interconnection weights from input to hidden layer and hidden layer to output layer respectively. c_k and d_k are translation and dilation weights respectively.

Gaussian wavelet activation functions is given by

$$\phi(x) = \frac{x}{\sqrt{2\pi}} exp(-x^2/2)$$

The common weight tuning algorithm is based on gradient descent algorithm. Let y_d be the desired NN output and y is the actual output. Then the backpropagated error is $E = y_d - y$. Now, the weights are updated to minimize the error E by the backpropagation algorithm given by:

$$v_{ij}^{(I+1)} = v_{ij}^{(I)} - \alpha_1 \frac{\partial E}{\partial v_{ij}^{(I)}} \tag{2.17}$$

$$w_{jk}^{(I+1)} = w_{jk}^{(I)} - \alpha_2 \frac{\partial E}{\partial w_{jk}^{(I)}} \tag{2.18}$$

$$c_k^{(I+1)} = c_k^{(I)} - \alpha_3 \frac{\partial E}{\partial c_k^{(I)}} \tag{2.19}$$

$$d_k^{(I+1)} = d_k^{(I)} - \alpha_4 \frac{\partial E}{\partial d_k^{(I)}} \tag{2.20}$$

where α_1, α_2, α_3 and α_4 are learning rates.

2.5 Robot Dynamics

Dynamic equation describes the relationships between the contact and actuation forces with the motion and acceleration trajectories that are due to the reacting forces. The precise knowledge of robot dynamics includes mass, location of center of mass and inertia of each link.

2.5.1 Lagrangian Equations

The Lagrangian L of a robotic system is the difference between the kinetic energy K and the potential energy P of the robot

$$L = K - P$$

The Lagrangian equations describing the motion of the system in terms of the generalized coordinates of joint variables $q = (q_1, q_2..., q_n) \in R^n$ and the joint torques/forces

$\tau = (\tau_1, \tau_2..., \tau_n)$ of the n-link manipulator are expressed as

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \;\; ; \;\; i = 1, 2, ..n$$

where$\dot{q}_i$ denotes the ith angular/linear velocity of ith actuator.

2.5.2 Dynamic Equations for 2 link Manipulator

In general dynamic equation of a n link manipulator is written as

$$M(q)\ddot{q}_s + C(q, \dot{q})\dot{q} + G(q) = \tau$$

where q_i are the joint angles, vector τ represents the applied torque/forces, M is inertia matrix, vector C represent the coriolis/centrifugal forces and G is the gravity vector.

According to Lagrange Euler method [20], the dynamic equation of the robot manipulator (given in Figure 2.1) are

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} (\frac{m_1}{3} + m_2)l_1^2 + m_2 C_2 l_1 l_2 + \frac{m_2 l_2^2}{3} & \frac{m_2 l_2^2}{3} + \frac{m_2 C_2 l_2^2}{2} \\ \frac{m_2 l_2^2}{3} + \frac{m_2 C_2 l_2^2}{2} & \frac{m_2 l_2^2}{3} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} m_2 S_2 l_1 l_2 \dot{\theta}_2^2 - m_2 S_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{1}{2} m_2 S_2 l_1 l_2 \dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} m_1 g l_1 C_1 + \frac{1}{2} m_2 g l_2 C_{12} + m_2 g l_1 C_1 \\ \frac{1}{2} m_2 g l_2 C_{12} \end{bmatrix}$$

where mass m_1 and m_2 are uniformly distributed, g is gravity and $C\theta_1 = \cos\theta_1$, $S\theta_1 = \sin\theta_1$, $C\theta_{12} = \cos(\theta_1 + \theta_2)$ and $S\theta_{12} = \sin(\theta_1 + \theta_2)$.

2.6 Biped Robot Configuration

Research on biped/humanoid robots is currently one of the most exciting topics in the field of robotics. In this section, we briefly discuss some basic concept related to biped robot.

2.6.1 Three Dimensional Motion

Let G be the universal frame with x-y-z axes as shown in Figure 2.5. Let the biped walks in x-direction. Then the three planes as shown in Figure 2.4 are:

1. **Frontal plane**: Parallel to the yz-plane.

2. **Sagittal plane**: Parallel to the xz-plane.

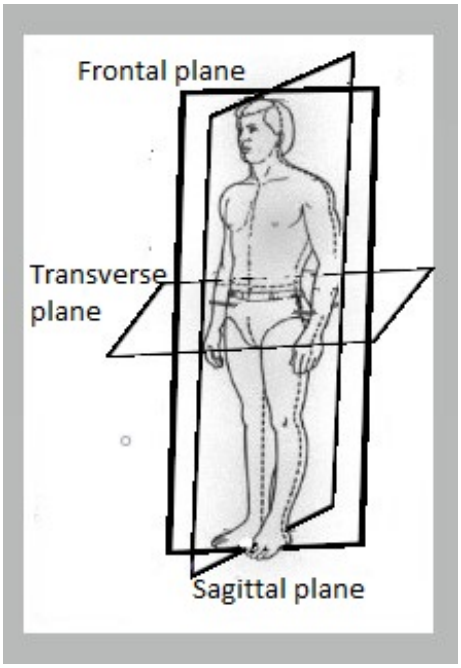


Figure 2.4: Coordinate frame

3. **Transverse plane**: Parallel to the xy-plane.

The motion ranges of the legs in the transverse plane and the frontal plane are negligible as compared to the motion in the sagittal plane. So, we consider the leg motion only in the sagittal plane.

2.6.2 Gait Cycle

Gait: Gait is the manner of walking. Amongst the gait patterns for different animals, human walking pattern is complicated. Different persons have different gaits. A gait cycle depends on various factors like terrain, speed and energy efficiency etc. A single gait cycle is also called as a step. A step (see Figure 2.5) can be divided into two phases:
(i) Single Support Phase(SSP): SSP refers to the duration of time in which stable leg's foot is on the ground and swing leg's foot is moving forward above the ground.
(ii) Double Support Phase(DSP): DSP refers to the situation in which both feet are touching on the ground.

Biped walking can be considered as a cyclic motion which is a repetition of one step. Throughout this paper **'stable foot'** means the foot which is on the ground during both

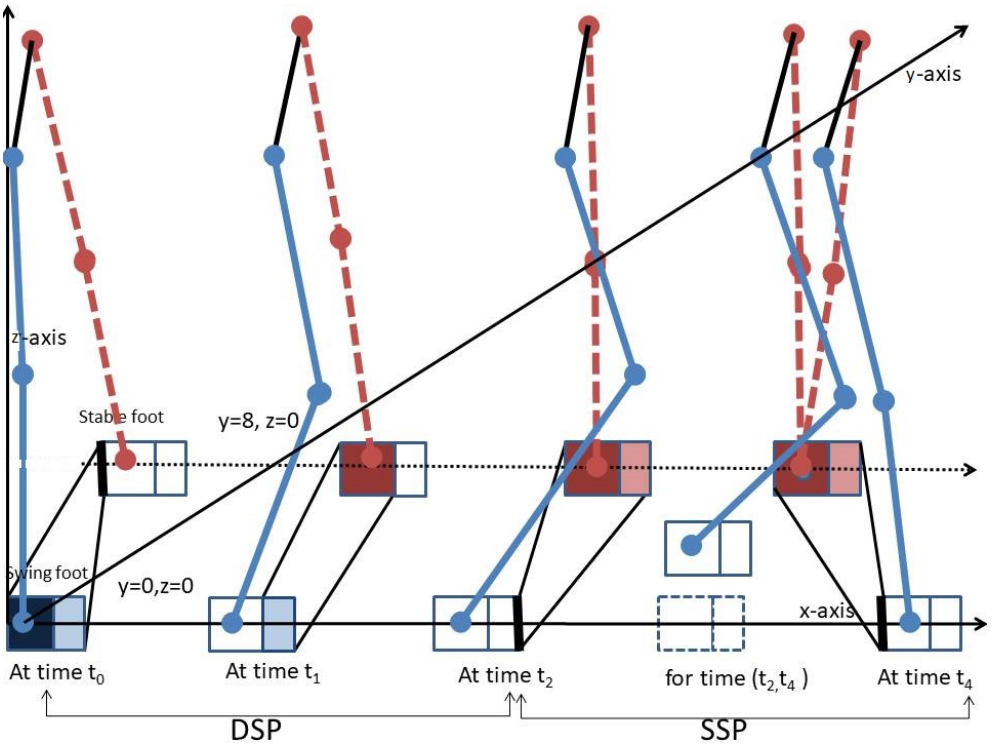


Figure 2.5: Swing and stable leg's gait

phases while **'swing foot'** means the one which moves forward above the ground in SSP.

2.6.3 Stability of Walk

Stability of a biped means the capability of the robot to maintain its body posture suitable for a given walking pattern/trajectory.

ZMP Stability

ZMP (Zero-Moment Point, ZMP) is defined as the point where the net moment of the inertial forces and the gravity forces along the axes parallel to the ground is equal to zero.

The x and y coordinates of ZMP are given by:

$$x_{ZMP} = \frac{\sum_{i=1}^{n} m_i(x_i(\ddot{z}_i + g) - \ddot{x}_i z_i)}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{2.21}$$

$$y_{ZMP} = \frac{\sum_{i=1}^{n} m_i(y_i(\ddot{z}_i + g) - \ddot{y}_i z_i)}{\sum_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{2.22}$$

where n is the number of links, m_i is the mass of the links and g is gravity.

Supported region/polygon: The convex hull of all floor contact points is called supported

region/polygon (Figure 2.6). Supported region is not fixed, it changes at every instant of time because of biped robot body movements, such as hip, upper body, toe and sole movements.

Stability margin: Stability margin for a given support polygon is the shortest distance between ZMP and the edges of the support polygon.

For SSP, the x-directional ZMP (x_{ZMP}) and the y-directional ZMP (y_{ZMP}) must lie inside

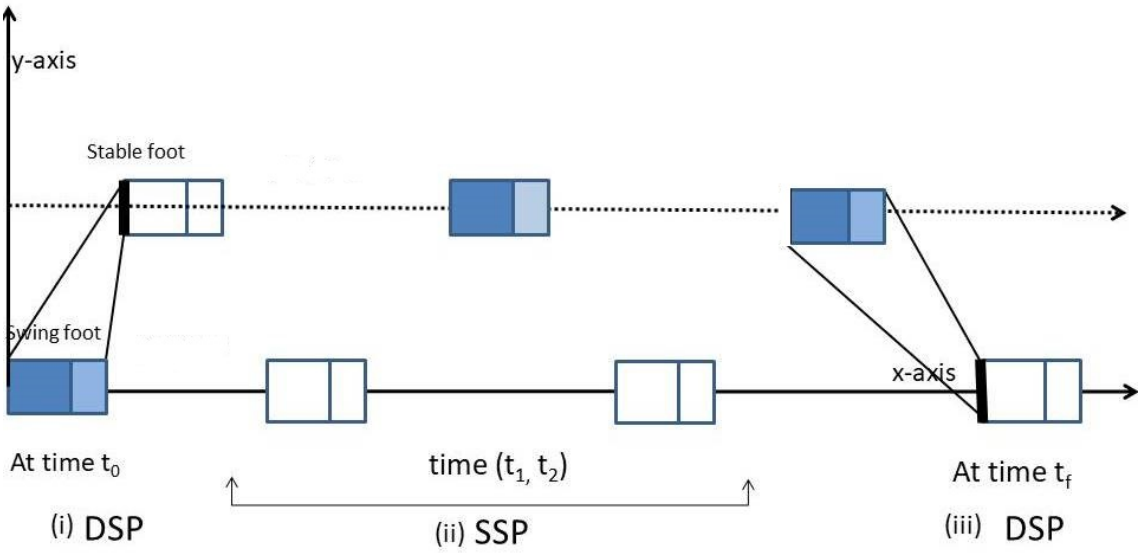


Figure 2.6: Supported region for SSP and DSP during one cycle(step) for flat footed robot

the stable foot region. For DSP, if the ZMP falls within the supporting polygon then biped robot can walk without falling down. The walk is more stable if ZMP lies strictly inside the supported region with larger stability margin. In typical human locomotion, the ZMP never stays in a fixed position rather moves forward in the direction of walk.

In Figure 2.6, the supported region (for SSP and DSP) at some time instant for flat footed robot is given. Blue shaded region shows that the foot is on the ground while light blue region represents the toe and dark blue region represents the sole of the foot. White region shows that the foot is above the ground. Grey shaded region between the two feet shows supported polygon.

2.6.4 Dynamic Equations of Biped Robot

Single Support Phase

During SSP, the swing leg leaves the contact surface and moves forward, and the whole body weight of robot is on the stable leg. The dynamical model during SSP can be written as

$$M_s(q_s)\ddot{q}_s + C_s(q_s, \dot{q}_s)\dot{q}_s + G_s(q_s) = \tau_s \tag{2.23}$$

where q_s are the joint angles, vector τ_s represents the applied torque/forces, M_s is inertia matrix, vector C_s represent the coriolis/centrifugal forces and G_s is the gravity vector.

Double Support Phase

The dynamical model of the instantaneous impact forces exerted by the foot and reaction torques at the joints at the time when the swing foot first touches the contact surface during the DSP can be written as

$$M_d(q_d)\ddot{q}_d + C_d(q_d, \dot{q}_d)\dot{q}_d + G_d(q_d) = \tau_d + F_{ext} \tag{2.24}$$

where F_{ext} represents the vector of torques acting at the joint of the biped.

Chapter 3: Stable NN Trajectory Tracking of Flat Footed Biped Robot with Upper Body Motion

Polynomial cubic spline trajectory is generated for ankle joints, hip joints and upper body so that the resulting walk is stable using Zero Momentum Point (ZMP) stability criteria with largest stability margin. Simulation are carried out using Matlab2010b.

3.1 Introduction

Trajectory planning is an important aspect to realize human like walk of a biped robot. The most widely used dynamic balance criterion is the ZMP. In human locomotion, the ZMP never stays in a fixed position, rather moves forward in the direction of locomotion [15, 17, 100, 101]. Narvez-Aroche et al. [67] presented the representative steps of the kinematic modeling and the computation of the ZMP. Huang et al. [31] proposed an iterative computation trajectory generation method for hip and foot by varying the values of the walking speed and step length to obtain the largest dynamic balance margin based on the ZMP. An approach is presented by Erbatur and Kurt [18] to improve the iterative computation trajectory generation of joints by taking into account a desired ZMP reference trajectory. In [18, 100], a cubic Hermitian polynomial interpolation algorithm is presented to implement on biped walk. Liu et al. [59] proposed a control, which is based on the motion of the upper body to maintain good stability of the biped and to relieve from knee bending problem.

Recently, much attention has been paid on FNN in robotics. FNN's capability and flexibility of learning by example, generalizing property, smaller information prerequisite,

quick real time operation and simplicity of implementation characteristics have made them useful in every field of research. NN can be used to find inverse kinematics solution for biped robot. In [5, 16, 17, 32]] researchers take some random samples of joint angles from specified range and used in forward kinematics equations to calculate the end effector's position and orientation for each set of joint angles. These pairs of joint angles and position/orientation are called training data. NN function is trained using supervised learning for given training data (approximately 1000) by giving position and orientation as inputs and the corresponding joint angles as desired outputs. Then NN (at least 100 neurons in hidden layer) gives inverse kinematics solutions for any position in workspace. De et al. [16] applied the above mentioned approach to find inverse kinematic solutions for humanoid robot. He used two NNs, one for DSP and second for SSP to learn the most stable positions from the range of all the possible solutions based on COM and no hip roll rotation.

In this chapter, first trajectories for ankle, hip and upper body are generated and then ZMP stability is analyzed. To ensure stability, three types of upper body motion are generated and whichever gives the best ZMP trajectory with largest stability margin is chosen. The inverse kinematics is solved using FNN with unsupervised learning procedure.

In Section 3.2, discussion on the robot model is given. Section 3.3 describes planning of leg trajectories for biped robot's walk with suitable conditions. Section 3.4 includes the forward kinematics and inverse kinematics of robot model. ZMP stability is calculated in Section 3.5. Upper body mass trajectories are discussed in Section 3.6. In Section 3.7, simulated results with graphs and discussion are presented. Conclusions are given in Section 3.8.

3.2 Robot Model

In Figure 3.1, each leg of biped robot have 2 degrees of freedom (DOF) with flat foot. All the joints are revolute which are called hip joint (H), knee joint (K) and ankle joint (A). Center of mass of upper body is denoted by (U). Total length of leg is $(l_1 + l_2)$ and length of foot is l_3. It is assumed that the length and mass of both the legs are same. The details of parameters are given in the Table 3.1.

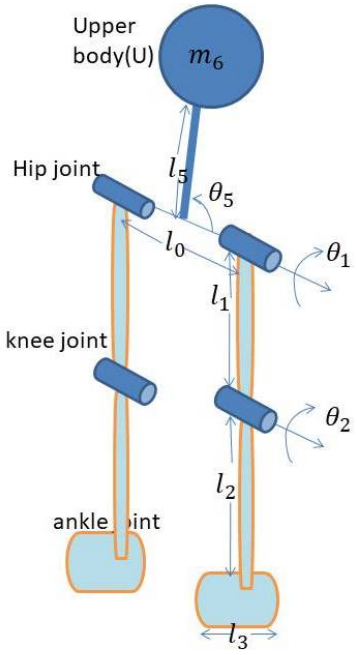


Figure 3.1: Schematic 3D biped model

Link	Length	Value	Mass	Value
HK	l_1	14 inches	m_1	4kg
KA	l_2	14 inches	m_2	4kg
HU	l_5	10 inches	m_6	50kg
HH	l_0	8 inches	m_5	4kg

Table 3.1: Parameters

Robot's walk can be considered as a repetition of one-step motion. The walking sequence can be determined by computing the trajectory of the hip, ankle and upper body joints. For hip trajectory, stable ankle joint is considered as a base and hip as the end effector, and for ankle trajectory, swing leg's hip is considered as base and its ankle joint as the end effector. Flat foot is attached at the ankle joint of each leg.

3.3 Trajectory Generation

Let the robot walk in sagittal plane (xz-plane).

3.3.1 Foot Trajectory

Let the swing leg's ankle joint follows a cubic polynomial trajectory. At time $t \in [t_0, t_f], t_0 = 0$, the ankle coordinate trajectories $(x_A(t), z_A(t))$ in x and z directions are computed by:

$$x_A(t) = a_1 + b_1 t + c_1 t^2 + d_1 t^3 \tag{3.1}$$

$$z_A(t) = l_1 + m_1 x_A(t) + n_1 x_A(t)^2 + p_1 x_A(t)^3 \tag{3.2}$$

with boundary conditions

$x_A(t_0) = x_0$; $x_A(t_f) = x_0 + x_f$; $\dot{x}_A(t_0) = 0$; $\dot{x}_A(t_f) = 0$.

$z_A(x_0) = 0$; $z_A(x_f) = 0$; $z_A(x_m) = h_1$; $\dot{z}_A(x_m) = 0$.

where h_1 is step height, x_0 is initial position, x_f is step length, total time is t_f and x_m is the x-coordinate at which maximum height is achieved.

Considering these boundary conditions, the x and z coordinates of ankle joint are given below:

$$x_A(t) = (x_0 + x_f) + \frac{(((3t_f^2 t_0 - t_f^3)(x_f))}{(t_f - t_0)^3} - \frac{(6t_f t_0(x_f))}{(t_f - t_0)^3} t + \frac{3(t_f + t_0)(x_f)}{(t_f - t_0)^3} t^2 - \frac{2(x_f)}{(t_f - t_0)^3} t^3 \tag{3.3}$$

$$z_A(t) = \frac{h(-(x_f + x_0)^2 x_0)}{(x_m - x_0)(x_m - x_f - x_0)^2} + \frac{h(x_f + x_0)(x_f + 3x_0) x_A(t)}{(x_m - x_0)(x_m - x_f - x_0)^2}$$

$$+ \frac{(-h(2x_f + 3x_0) x_A(t)^2 + h x_A(t)^3)}{(x_m - x_0)(x_m - x_f - x_0)^2} \tag{3.4}$$

3.3.2 Hip Trajectory

For biped robot walking on a plane, motion of the stable leg is assumed to be like an inverted pendulum considering it's ankle joint as base and hip as end effector. While walking, humans do not fold their stable leg as the whole body weight lies on it. Let the hip follows a circular path with center at ankle joint A and radius $(l_1 + l_2)$ with suitable boundary conditions. At time $t \in [t_0, t_f], t_0 = 0$, the hip coordinates trajectories $(x_H(t), z_H(t))$ in xz plane are assumed to be:

$$x_H(t) = q_0 + q_1 t + q_2 t^2 + q_3 t^3 \tag{3.5}$$

$$z_H(t) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_0 + x_f/2))^2} \tag{3.6}$$

with boundary conditions

$x_H(t_0) = x_0 + x_f/4$; $x_H(t_f) = x_0 + 3x_f/4$; $\dot{x}_H(t_0) = v_s$; $\dot{x}_H(t_f) = v_e$.

$z_H(t_0) = h$; $z_H(t_f) = h$; $\dot{z}_H(t_0) = v_{zs}$; $\dot{z}_H(t_f) = v_{ze}$.

where h is maximum hip height at time t_2, h_0 is hip height at both starting and end positions.

Hence, hip trajectory in x and z directions are given by:

$$x_H(t) = x_0 + \frac{x_f}{4} + v_s t + \left(\frac{(v_e - v_s)}{2t_f} - r_4\frac{3t_f}{2}\right)t^2 - 2\left(\frac{x_f}{2t_f^3} - \frac{(v_s + v_e)}{2t_f^2}\right)t^3 \quad (3.7)$$

$$z_H(t) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_0 + x_f/2))^2} \quad (3.8)$$

where $r_4 = -2\left(\frac{x_f}{2t_f^3} - \frac{(v_s+v_e)}{2t_f^2}\right)$

3.4 Forward and Inverse Kinematics

The kinematic equations of swing leg's ankle is obtained by considering hip joint (H) as a base and ankle joint (A) as the end effector. So, the forward kinematic equations of the swing leg are

$$x_A(t) - x_H(t) = l_1 cos\theta_1(t) + l_2 cos(\theta_1(t) + \theta_2(t)) \quad (3.9)$$

$$z_A(t) - z_H(t) = l_1 sin\theta_1(t) + l_2 sin(\theta_1(t) + \theta_2(t)) \quad (3.10)$$

where $(x_A(t), z_A(t))$ and $(x_H(t), z_H(t))$ are defined as earlier and and θ_1, θ_2 are joint angles as shown in Figure 3.1.

Stable leg's ankle joint is fixed on the ground and knee joint is locked (no rotation) while hip is moving. Thus, the stable leg moves like single link manipulator with A as base and H as end effector. Its forward kinematic equations are

$$x_H(t) - \left(x_0 + \frac{x_f}{2}\right) = (l_1 + l_2)cos\theta_3(t) \quad (3.11)$$

$$z_H(t) = (l_1 + l_2)sin\theta_3(t) \quad (3.12)$$

where $\left(x_0 + \frac{x_f}{2}, l_0, 0\right)$ is the position of the stable leg's ankle joint which lies on the plane $y = l_0$.

3.4.1 FNN Learning Scheme for Inverse Kinematics

In this section, the inverse kinematic problem is solved using FNN. Analytic solutions for the inverse kinematic problem of a 2-link manipulator can be obtained very easily. However there are two solutions (joint angles) for a given end effector position. So, when the end effector is moving continuously, it becomes very difficult to choose one solution at every instant of time so that the resulting inverse kinematic solutions are continuous. This difficultly is removed if FNN is used to solve the inverse kinematics. At a particular instant of time if the initial weights to the FNN are the final weights of the previous instant, then we get the desired continuous output.

Proposed FNN Method

A two-layer FNN(as in Section 2.3) with M input units, m output units and N units in the hidden layer, is shown in the Figure 3.2.

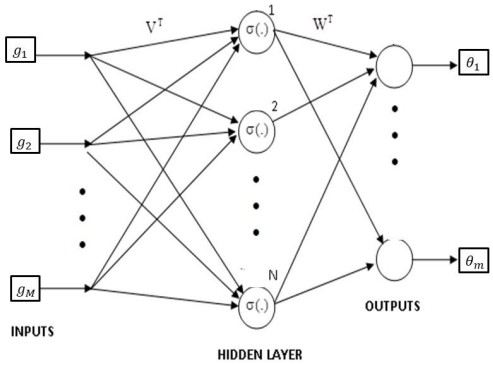


Figure 3.2: Feedforward neural network

Further, the output vector θ is determined in terms of the input vector g by the formula

$$\theta_i = \sum_{j=1}^{N} \left[w_{ij} \sigma \left(\sum_{k=1}^{M} v_{jk} g_i \right) \right]; \quad i = 1, 2, ..., m \tag{3.13}$$

where $\sigma(.)$ is the activation function of the neurons of the hidden layer. The input to hidden layer interconnection weights are denoted by v_{jk} and the hidden layer to output interconnection weights by w_{ij}.

There are many classes of activation functions e.g. sigmoid, tangent hyperbolic and Gaussian. The sigmoid activation function used in this chapter given by

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.14}$$

By collecting all the FNN weights v_{jk} and w_{ij} into matrices of weights V=$[v_{jk}]_{\substack{j=1,2...N \\ k=1,2...M}}$ and W=$[w_{ij}]_{\substack{i=1,2...m \\ j=1,2...N}}$ respectively, FNN equation 3.13 can be written in terms of vectors as

$$\theta = W\sigma(Vg_{de}) \tag{3.15}$$

Now, let us consider the processor for solving inverse kinematic solutions for the ankle trajectory whose forward kinematic equations are given by (3.9), (3.10). $g_{de} = (x_{de}(t), z_{de}(t)) = (x_A(t) - x_H(t), z_A(t) - z_H(t))$ are the x- and z-position coordinates to follow ankle trajectory treating hip as base for each $t\epsilon[0, t_f]$. These coordinates become inputs in neural network (3.13) to get NN joint angles $\theta_{nn}(t)$ as outputs (see Figure 3.2). By substituting these $\theta_{nn}(t)$ values in the forward kinematic equations the values which are obtained, called NN position coordinates $(x_{nn}(t), y_{nn}(t))$. The desired and NN position coordinates are compared and the resulting error is given by

$$E(t) = (x_{de}(t) - x_{nn}(t))^2 + (z_{de}(t) - z_{nn}(t))^2 \tag{3.16}$$

is minimized by updating weights.

The weights are updated by the learning rule given by:

$$W_{n+1}(t) = W_n(t) - \alpha\delta \quad 0 < \alpha < 1 \tag{3.17}$$

$$V_{n+1}(t) = V_n(t) - \alpha\eta \tag{3.18}$$

where,

$$\delta = \frac{\partial E}{\partial w_{ij}} \;\; i = [1:2], \;\; j = [1:10] \tag{3.19}$$

$$\eta = \frac{\partial E}{\partial v_{jk}} \;\; j = [1:10], \;\; k = [1:2] \tag{3.20}$$

Here, $W_n(t)$ and $V_n(t)$ are the weights matrices at the nth iteration. n and α represents the iteration number and learning rate respectively. This process is repeated until the absolute error become less than the tolerance value, so that the desired joint angles are obtained.

3.5 Stability of Walk and Upper Body Motion

3.5.1 ZMP Stability Analysis

Stability is ensured by using ZMP ceriteria as mentioned in Section 2.6.3 of Chapter 2.

For stable walking, ZMP should lies inside the stable foot region in single support phase and it should lies within the supported region in double support phase. Modern walking robots usually have heavy upper body due to batteries and electronic circuits and which affects the stability. To ensure stable walking, ZMP must be within the supported region. For this, the parameters of upper body are changed to figure out the suitable ZMP trajectory which moves in a desired manner.

3.5.2 Upper Body Motion on the Frontal Plane

The total mass of upper body is assumed to be a single mass point at COM (of the upper body) for planning its trajectory.

On the frontal plane, upper body mass shifts from one position to another (only parallel to y-direction) and its trajectory in y-direction highly affects the y-ZMP trajectory. In order to find a desirable ZMP trajectory, three cases of upper body mass trajectory in y-direction are generated and the one which ensures the higher stability margin is chosen. These trajectories are determined by cubic spline and detailed discussion is given below:

Case-1: As the robot starts its step, upper body starts to move from the middle of the hips towards the stable leg's hip during the time interval $[t_0, t_1]$, stays there during the time interval $[t_1, t_3]$, then return back to the middle of the hips during time interval $[t_3, t_f]$ in y-direction where $t_i = it_f/4$, $i = 0, 1, 2, 3, 4$. So, the upper body trajectory in y-direction is:

$$y_M(t) = \begin{cases} y_l + v_{0y}t + (\frac{3(y_a-y_l)}{{t_1}^2} - \frac{2v_{0y}}{t_1})t^2 + (\frac{-2(y_a-y_l)}{{t_1}^3} - \frac{y_v}{{t_1}^2})t^3 & t_0 \le t \le t_1 \\ y_a & t_1 \le t \le t_3 \\ (y_a + \frac{(-3t_f t_3^2+t_3^3)(y_l-y_a)}{(t_3-t_f)^3} + \frac{t_f t_3^2 y_v}{(t_3-t_f)^2}) & \\ (\frac{6t_f t_3(y_l-y_a)}{(t_3-t_f)^3} - \frac{(t_3^2+2t_f t_3)y_v}{(t_3-t_f)^2})t + (\frac{-3((y_l-y_a)(t_3+t_f)}{(t_3-t_f)^3} & \\ +\frac{y_v(4t_3+2t_f)}{2(t_3-t_f)^2})t^2 + (\frac{2(y_l-y_a)}{(t_3-t_f)^3} - \frac{y_v}{(t_3-t_f)^2})t^3 & t_3 \le t \le t_f \end{cases}$$

where y_l is the middle point of the hips, y_a is the extreme position of the upper body

mass, and $v_{0y} > 0$ is initial velocity of upper body mass.

Case-2: As robot starts its step, upper body starts to move from middle of hips to the side of the stable leg's hip within time t_0 to $t_f/8$, stay there for time $(t_f/8, 7t_f/8)$ and then return back towards middle of hips within time $7t_f/8$ to t_f in y-direction. Then, the upper body trajectory can be calculated by equation 3.5.2 by putting $t_1 = t_f/8$ and $t_3 = 7t_f/8$.

Case-3: As the robot start its step, upper body start to move from middle of hips to the side of the stable leg's hip within time t_0 to t_2, then return back towards middle of hips within time t_2 to t_f in y-direction. So the upper body trajectory in y-direction is:

$$y_M(t) = \begin{cases} y_l + y_v t + (\frac{3(y_a - y_l)}{{t_2}^2} - \frac{2y_v}{t_2})t^2 + (\frac{-2(y_a - y_l)}{{t_2}^3} - \frac{y_v}{{t_2}^2})t^3 & t_0 \le t \le t_2 \\ (y_a + \frac{(-3t_f t_2^2 + t_2^3)(y_l - y_a)}{(t_2 - t_f)^3} + \frac{t_f t_2^2 y_v}{(t_2 - t_f)^2}) & \\ (\frac{6t_f t_2 (y_l - y_a)}{(t_2 - t_f)^3} - \frac{(t_2^2 + 2t_f t_2) y_v}{(t_2 - t_f)^2})t + (\frac{-3((y_l - y_a)(t_2 + t_f)}{(t_2 - t_f)^3} & \\ + \frac{y_v(4t_2 + 2t_f)}{2(t_2 - t_f)^2})t^2 + (\frac{2(y_l - y_a)}{(t_2 - t_f)^3} - \frac{y_v}{(t_2 - t_f)^2})t^3 & t_2 \le t \le t_f \end{cases}$$

3.6 Results

Let total length of foot is 6 units and width is 4 units, initial and end velocity for ankle is 0 unit/sec. Ankle is fixed at the middle point of the foot, so that the initial x coordinate of the ankle is $x_0 = 3$ units. The ankle joint covers a step length x_f=14 units from initial position $(x_0, 0, 0)$ to the final position $(x_0 + x_f, 0, 0)$ with step height h=2.5 units. At any given instant of time the middle point of the hips is y_l=4 units, extreme y coordinate of the upper body mass is y_a=8.5 units. Initially, swing foot lies on the xy plane in the region $0 < x < 6$ units and $-2 < y < 2$ units and stable foot lies on the xy plane in the region $7 < x < 13$ units and $6 < y < 10$ units.

Figure 3.3 represents the desired trajectory graph for the ankle and hip. The inverse kinematics of these trajectories are calculated using FNN in Matlab2010b. Figures 3.4(a) and 3.4(b) shows the inverse kinematic solutions using FNN of swing leg and stable leg respectively with respect to time. Figures 3.5(a) and 3.5(b) show that the desired trajectories are followed by the swing leg and stable leg for one step using the inverse kinematic solutions.

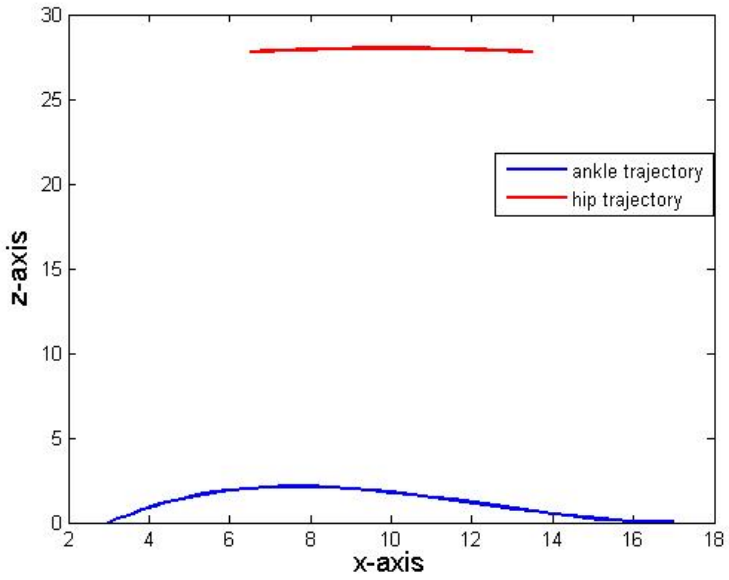
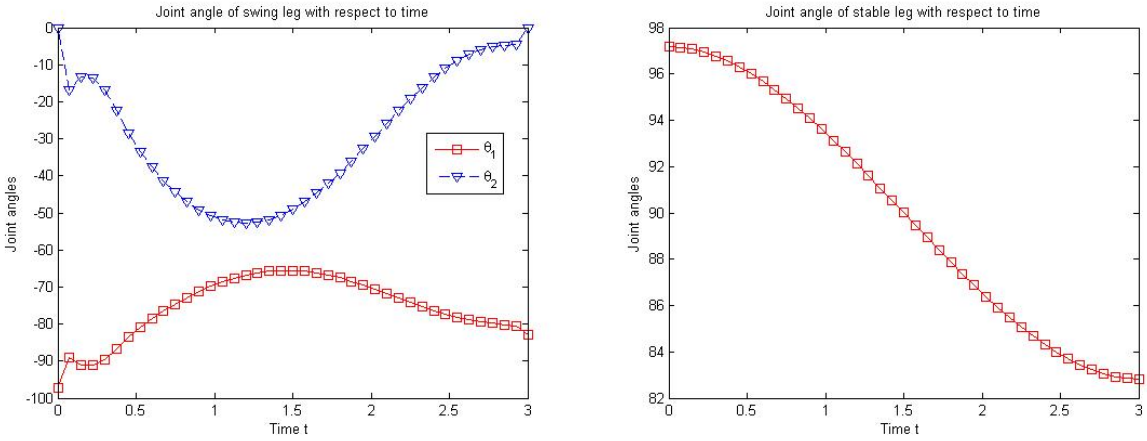


Figure 3.3: Ankle and Hip trajectory in xz-plane

Figure 3.6(a) describes the error between the position calculated by NN and actual position of the swing leg's ankle joint (vertical axis) versus the neural network iteration number(horizontal axis). Similarly, Figure 3.6(b) describes the error between the position calculated by NN and actual position of the stable leg's hip(vertical axis) versus the neural network iteration number(horizontal axis). Each curve in Figures 3.6(a), 3.6(b) represents the error graph at a time instant $t = 0.1i$, where $i = 0, 1, 2, ..., 40$.

Figures 3.7, 3.8 and 3.9 represent the motion of the ZMP of this biped in the xy plane for three cases of upper body motion. Here, the step time t_f is taken to be 3 secs. As we can see from these figures, the ZMP trajectories for Case-1 and Case-3 are inside the support polygon and hence provide stable walking for the desired ankle and hip trajectories. For step time t_f= 2 secs, Case-1 gives marginally stable walk while Case-3 ensured stable walking. For step time t_f= 1.5 secs, Case-1 provides unstable walk(Figure 3.7(b)) while Case-3 provides stable walk (Figure 3.9). So, this experiment results that the upper body motion given in Case-3 is suitable for stable walking of the biped robot to track the desired ankle and hip trajectories as given in Table 3.2. Upper body motion given in Case-2 provides the unstable walk in all the cases.

Whole body motion in 3D for one step with step time $t_f = 1.5$ for Case-3 of upper



(a) For swing leg θ_1 and θ_2

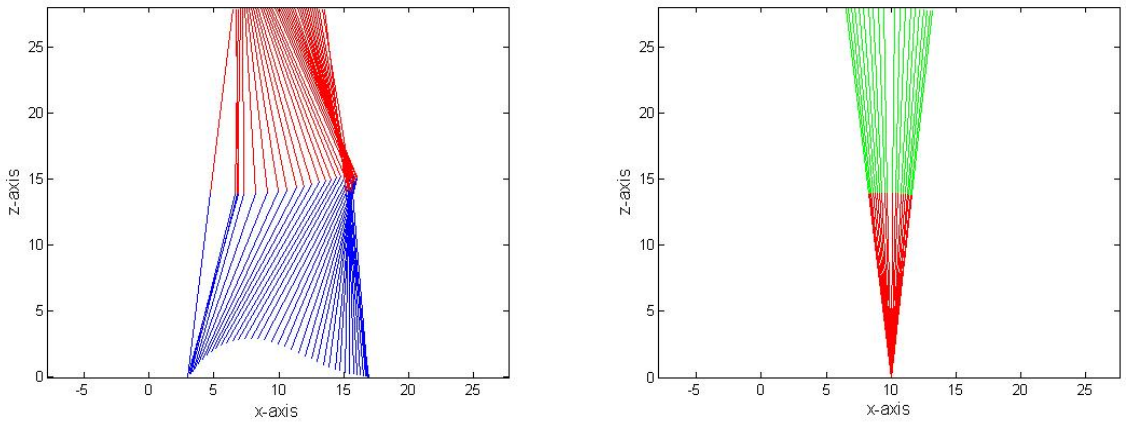
(b) For stable leg θ_3

Figure 3.4: Inverse kinematics solution using FNN $t\epsilon[0, t_f]$

body trajectory is given in Figure 3.10.

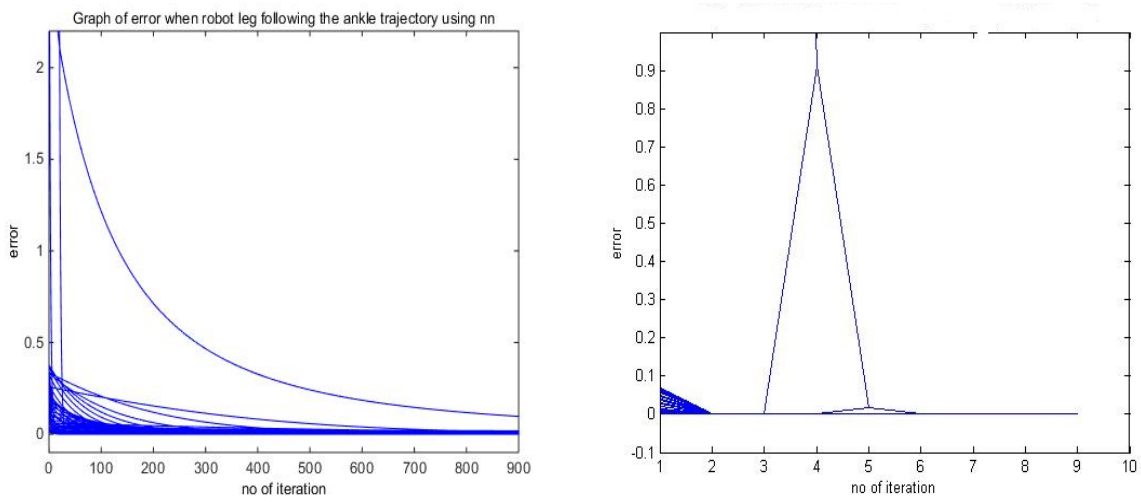
3.7 Conclusion

To ensure stability, three cases of upper body motion are considered and it is concluded from the simulation results that Case-3 of upper body motion gives the best ZMP trajectory with largest stability margin. In this chapter, the biped robot with flat feet is considered. So the above observation is suitable, specifically for the flat footed robots but this conclusion may not be applicable for toe footed robots as we will see in the subsequent chapter.



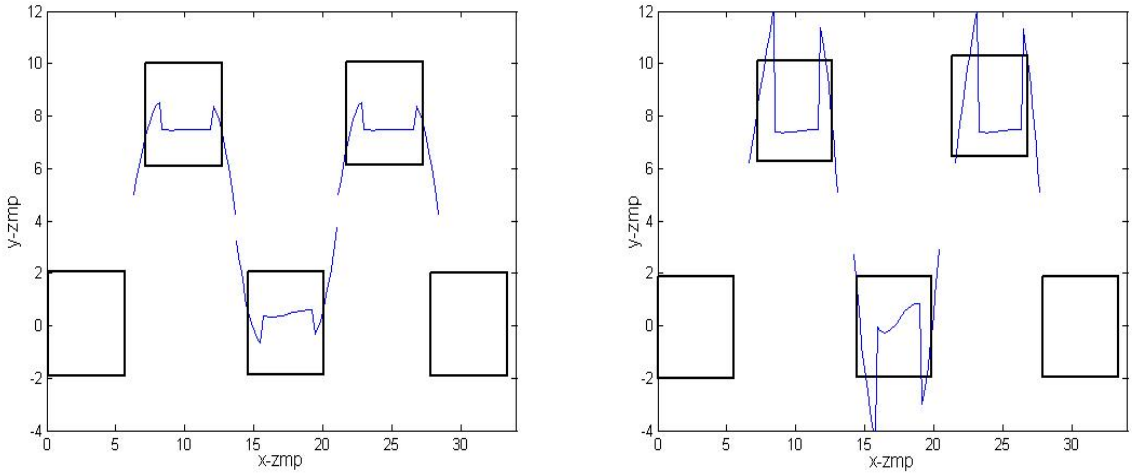
(a) Swing leg (b) Stable leg

Figure 3.5: Following the given trajectories using FNN



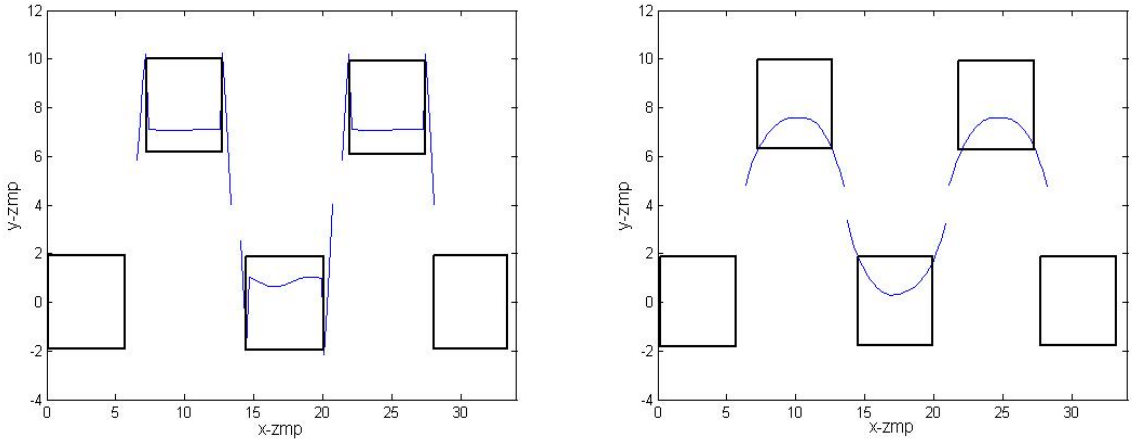
(a) For swing leg (b) For stable leg

Figure 3.6: Error convergance graphs for inverse kinematics solutions using FNN



(a) in 3 secs (b) in 1.5 secs

Figure 3.7: Stable ZMP trajectory of Case-1



(a) Case-2 (b) Case-3

Figure 3.8: ZMP trajectories in 3 secs

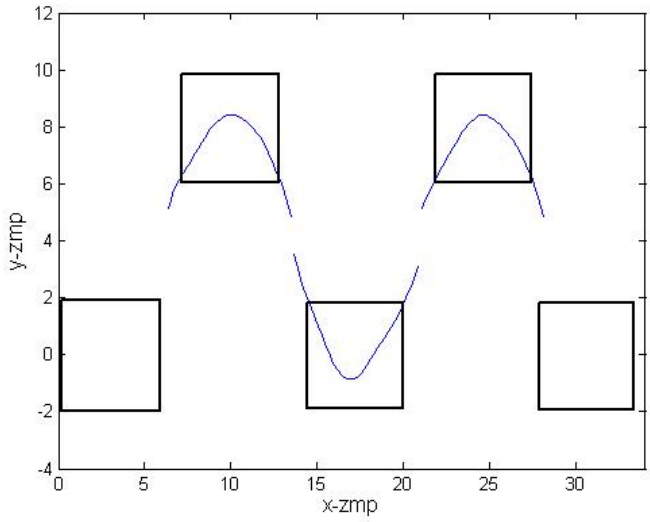


Figure 3.9: Stable ZMP trajectory in 1.5 sec for Case-3

Hip		**Upper Body**		ZMP
velocity	Time	Trajectory	initial velocity	stability
v_s=2.3in/s	3s	Case-1	$v_{y0} = 10in/s$	stable
v_s=3.5in/s	2s	Case-1	$v_{y0} = 15in/s$	stable but small margin
v_s=4.7in/s	1.5s	Case-1	$v_{y0} = 20in/s$	unstable
v_s=2.4in/s	3s	Case-2	$v_{y0} = 16in/s$	unstable
v_s=3.5in/s	2s	Case-2	$v_{y0} = 20in/s$	unstable
v_s=4.7in/s	1.5s	Case-2	$v_{y0} = 22in/s$	unstable
v_s=2.3in/s	3s	Case-3	$v_{y0} = 7.3in/s$	stable
v_s=3.5in/s	2s	Case-3	$v_{y0} = 10.3in/s$	stable
v_s=4.7in/s	1.5s	Case-3	$v_{y0} = 11in/s$	stable

Table 3.2: Comparison Table

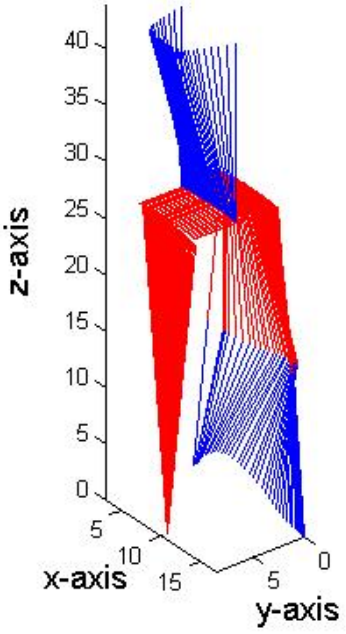


Figure 3.10: Biped walk in 3D for one step

Chapter 4: Toe Footed Biped Robot Model for Stable Human like Gait with Upper Body Motion

A polynomial approach for trajectory generation is presented for biped robot which can walk like human with movable foot and active toe. The proposed approach allows smooth transition between the walking phases namely, single and double support phase. Zero Moment Point (ZMP) stability is ensured by taking into account the upper-body movements along with the planned motion trajectories.

4.1 Introduction

To achieve human gait it is important to plan suitable trajectories for the robot joints during the walk. Xiaoguang and Ruyi [100] presented a humanoid robot gait planning. The authors in [15, 94, 100] presented a cubic polynomial interpolation approach to implement biped walk. In [15, 51, 90, 104], trajectory generation for biped walking on different types of surfaces have been considered. Humans tend to bend their foot on toe to avoid bending of knee while walking in a comfortable way. Since 2000 [48, 68], toe footed biped robots are being constructed and different kinds of designs are presented. In [67], the authors used toe joint rotation during the double support phase. The addition of toe on foot is required for implementing the natural way of walking as compared to flat foot. The knee bending problem can be relieved by allowing a vertical body motion (VBM) [59, 88].

Liu et al. [59] proposed a controller, which is based on the motion of the upper body, to maintain good stability of the biped robot and to avoid knee bending. Xu et al. [101] studied ZMP stability by adjusting the hip trajectory using cubic spline function. In this chapter, a

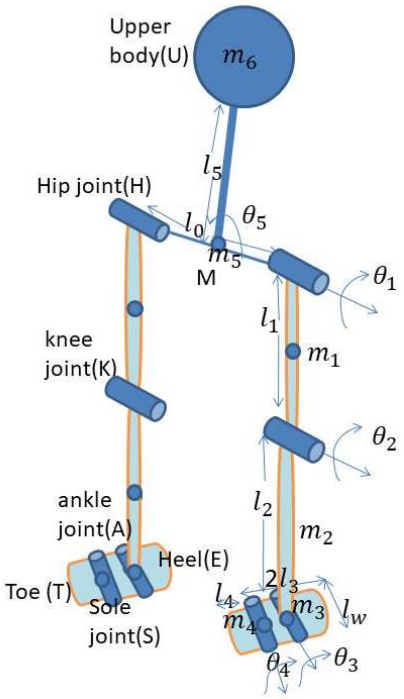


Figure 4.1: Robot Structure

polynomial trajectory algorithm is given to provide a simple and effective walking pattern. Ankle and sole joint rotations are considered in DSP for smooth transition between the walking phases. Three different cases for upper body trajectories and ankle trajectories are considered and the effect of these trajectories on ZMP stability is analyzed.

This chapter is organized as follows. Section 4.2 describes the robot model. In Section 4.3, the joint trajectories are planned with suitable boundary conditions. ZMP stability analysis and upper body trajectories are discussed in Section 4.4. Section 4.5 includes simulation results with graphs and discussions. Conclusion is presented in Section 4.6.

4.2 Robot Model

In Figure 4.1, each leg of the biped robot has 4 degrees of freedom(DOF) and the upper body has one DOF. Hip joint, Knee joint, Ankle joint of swing leg and stable leg are represented by H, K, A and Hs, Ks, As respectively. All the joints are revolute joints.

Let center of mass of upper body is denoted by (U). The tip of the toe is T and heel of the foot is E for swing leg and the tip of the toe is Ts and heel of the foot is Es for stable

Link	Length	value	mass	value
HK	l_1	14 units	m_1	4 units
KA	l_2	14 units	m_2	4 units
ES	$2l_3$	4 units	m_3	0.8 units
ST	l_4	2 units	m_4	0.2 units
HU	l_5	12 units	m_6	50 units
HH	l_0	8 units	m_5	4 units
foot width	l_w	3 units		

Table 4.1: Description of robot links

leg. Further, it is assured that the length and mass of both the legs are same and the details of the parameters are given in the Table 4.1.

The motion of legs in sagittal plane is much greater compare to its motion in the frontal and transverse planes, therefore, the effects of these motions are negligible. Hence it is assumed that the robot is walking in sagittal plane. A single cycle of walk for biped robot can be divided into two parts: legs motion and upper-body motion (see Figure 4.2). Motion of legs maintains the gait while the upper body motion is for the stability.

The time interval of single cycle(one step) of walk is assumed as (t_0, t_4) where $t_0 = 0$. For computing the joint trajectories for one step, the total time is divided into four equal intervals $(t_{i-1}, t_i),\ \ i = 1, 2, 3, 4$ where $t_i = iT_g$ and $T_g = t_4/4$. The trajectory planning of all joints in cartesian plane are divided into two phases; Phase-1 is associated to DSP during (t_0, t_2) and Phase-2 is associated to SSP during (t_2, t_4). x_{f1} is total length travelled by the heel and $x_f = x_{f1} - l_3(1 - \cos\theta_6)$ is the total step length traveled by the ankle (see Figure 4.3(b)).

4.3 Trajectory Planning

4.3.1 Swing Leg's Joint Trajectories

At $t = 0$, the coordinates of E, A, S, and T respectively are (0,0,0), $(l_3, 0, 0)$, $(2l_3, 0, 0)$, $(2l_3{+}l_4, 0, 0)$. Let the (x,y,z) coordinates of the joint A during Phase:i=1, 2 be $(x_{A_i}(t), 0, z_{A_i}(t))$. The coordinates of other joints are denoted in similar way by replacing A with a suitable notation. Initially, the foot TSE of the swing leg is on the ground as a 2 link manipulator with first link length TS=l_4, second link length SE=$2l_3$.

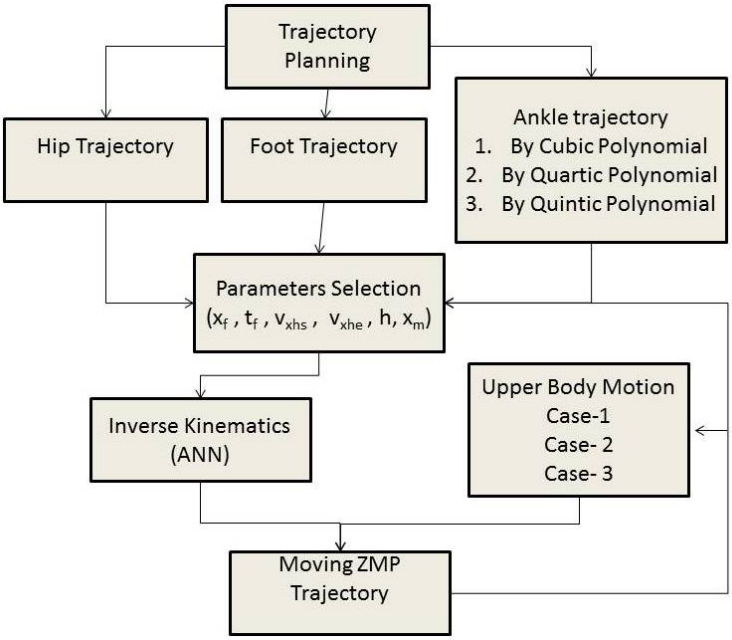


Figure 4.2: Algorithm

Phase-1: During the time interval $[0, t_1]$(see Figure 4.3(a)), the ankle joint A follows a circular path with center at S and radius l_3 while the points S and T are stationary at $(2l_3, 0, 0)$ and $(2l_3 + l_4, 0, 0)$ respectively. In Figure 4.3(a), it is shown that the sole SE subtends an angle $\theta_{3S} = 0$ at $t = 0$ and $\theta_{3S} = -\theta_a$ at $t = t_1$ with the x-axis. Here, θ_a is a designed parameter.

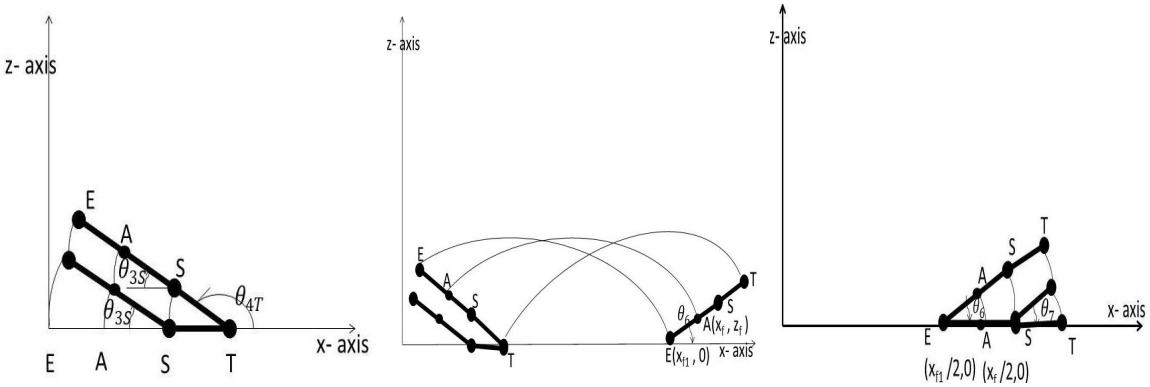
During the time interval $[t_1, t_2]$, the point S moves in a circular path with center at T and the radius l_4 while the point T remains stationary at $(2l_3+l_4, 0, 0)$. Figure 4.3(b) shows that the toe TS subtends an angle $\theta_{4T} = \pi$ at t=t_1 and $\theta_{4T} = \pi - \theta_b$ at $t = t_2$ with the x-axis where θ_b is another designed parameter. Simultaneously, the sole SE reverse its previous circular motion. Now, smooth trajectories are designed for the joint angles, θ_{3S} and θ_{4T} during Phase-1 using above boundary conditions (as given in Table-4.2).

The forward kinematic equations of AT during DSP $(0, t_2)$ is considered as a 2-link (TS and SA) manipulator treating toe tip T as base and ankle joint A as end effector are

$$l_4 cos\theta_{4T}(t) + l_3 cos(\theta_{3S}(t) + \theta_{4T}(t)) = x_A(t) - (2l_3 + l_4) \tag{4.1}$$

$$l_4 sin\theta_{4T}(t) + l_3 sin(\theta_{3S}(t) + \theta_{4T}(t)) = z_A(t) \tag{4.2}$$

Similarly, The forward kinematic equations for $(x_E(t), 0, z_E(t))$ and $(x_S(t), 0, z_S(t))$ are:



(a) Swing foot during DSP (b) Swing foot during SSP (c) Stable foot during DSP

Figure 4.3: Feet movement during DSP and SSP

$$(l_4 cos\theta_{4T}(t) + 2l_3 cos(\theta_{3S}(t) + \theta_{4T}(t)) = x_A(t) - (2l_3 + l_4) \tag{4.3}$$

$$l_4 sin\theta_{4T}(t) + 2l_3 sin(\theta_{3S}(t) + \theta_{4T}(t)) = z_A(t) \tag{4.4}$$

$$(l_4 cos\theta_{4T}(t) + 2l_3 cos(\theta_{3S}(t) + \theta_{4T}(t)) = x_A(t) - (2l_3 + l_4) \tag{4.5}$$

$$l_4 sin\theta_{4T}(t) + 2l_3 sin(\theta_{3S}(t) + \theta_{4T}(t)) = z_A(t) \tag{4.6}$$

Phase-2: In this phase, ankle trajectory is computed by fitting a polynomial for given constraints from time t_2 (when foot leaves the ground) to final time t_4 (where again the foot touches the ground).

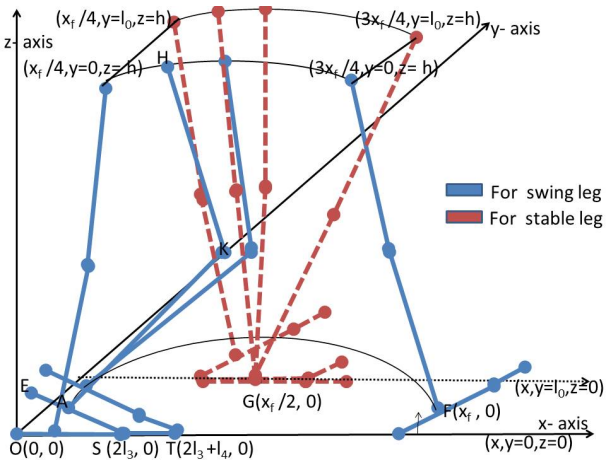
Three types of polynomials: cubic, quartic and quantic are considered for ankle trajectory. Boundary conditions for these trajectories are provided in Phase-2 of Table-4.2. The coefficients of these polynomials are calculated by using polynomial interpolation.

Polynomial interpolation approach To satisfy four boundary conditions as provided in Table 4.2, the cubic polynomial is considered as follows:

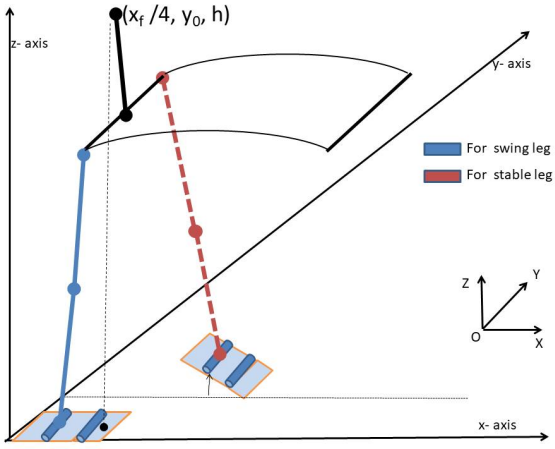
$$x_{A_2}(t) = n_0 + n_1 t + n_2 t^2 + n_3 t^3 \tag{4.7}$$

$$\frac{dx_{A_2}(t)}{dt} = n_1 + 2n_2 t + 3n_3 t^2 \tag{4.8}$$

By substituting the boundary conditions in equations 4.7 and 4.8, we get



(a) Walk parameters



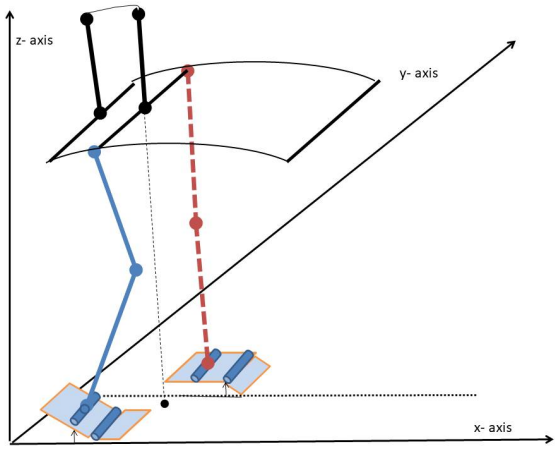
(b) at time t_0

$$n_0 + n_1 t_2 + n_2 t_2^2 + n_3 t_2^3 = x_e \tag{4.9}$$

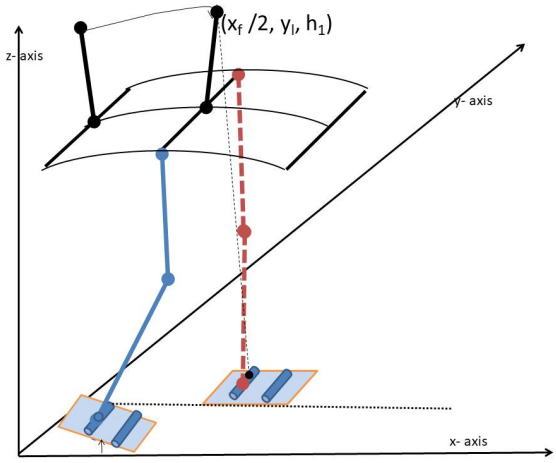
$$n_0 + n_1 t_4 + n_2 t_4^2 + n_3 t_4^3 = x_f \tag{4.10}$$

$$n_1 + 2n_2 t_2 + 3n_3 t_2^2 = x_v \tag{4.11}$$

$$n_1 + 2n_2 t_4 + 3n_3 t_4^2 = 0 \tag{4.12}$$



(c) at time t_1

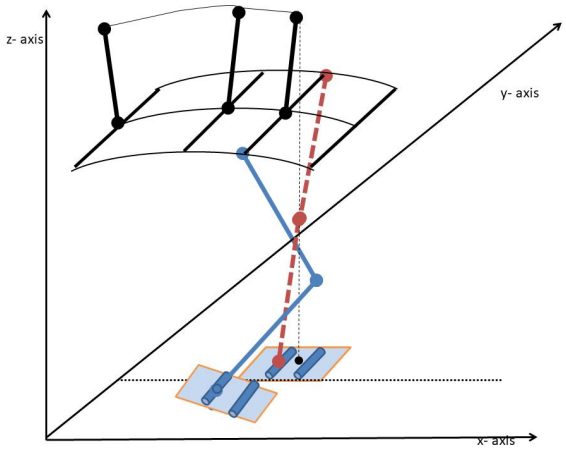


(d) at time t_2

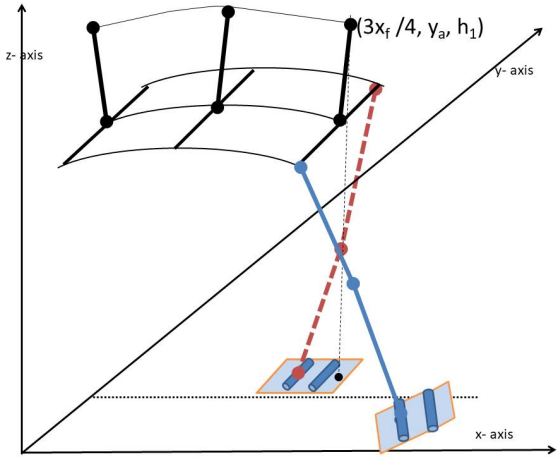
The matrix representation for these equations 4.9-4.12 is

$$\begin{bmatrix} x_e \\ x_f \\ x_v \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_4 & t_4^2 & t_4^3 \\ 0 & 1 & 2t_2 & 3t_2^2 \\ 0 & 1 & 2t_4 & 3t_4^2 \end{bmatrix} \begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

Then, the coefficients of the polynomial can be calculated by



(e) at time t_3



(f) at time t_4

Figure 4.4: Different phases of biped robot's walk

$$\begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_4 & t_4^2 & t_4^3 \\ 0 & 1 & 2t_2 & 3t_2^2 \\ 0 & 1 & 2t_4 & 3t_4^2 \end{bmatrix}^{-1} \begin{bmatrix} x_e \\ x_f \\ x_v \\ 0 \end{bmatrix}$$

which is denoted by

$$N_{4\times1} = A_{4\times4}^{-1}.M_{4\times1} \tag{4.13}$$

In general, for n boundary conditions, a polynomial of degree $n-1$ can be generated and

its coefficients can be obtained by:

$$N_{n\times 1} = A^{-1}_{n\times n}.M_{n\times 1} \tag{4.14}$$

Similarly, the coefficients of quartic and quantic polynomials can be obtained.

	Trajectories for Swing leg		
	Phase-1 $(t \in (t_0, t_2))$		
	Boundary conditions	**Trajectory**	**Coefficients**
	$\theta_{3S}(t_0) = 0,\ \theta_{3S}(t_1) = -\theta_a,$ $\dot{\theta}_{3S}(t_0) = 0,\ \dot{\theta}_{3S}(t_1) = 0,$ $\theta_{3S}(t_1) = -\theta_a,\ \theta_{3S}(t_2) = 0,$ $\dot{\theta}_{3S}(t_1) = 0,\ \dot{\theta}_{3S}(t_2) = 0,$	$\theta_{3S}(t) = p_1 + p_2t + p_3t^2 + p_4t^3$ $t \in (t_0, t_1)$ $\theta_{3S}(t) = q_1 + q_2t + q_3t^2 + q_4t^3$ $t \in (t_1, t_2)$	$p_1 = 0 \quad p_2 = 0$ $p_3 = \frac{3\theta_a}{T_g^2} \quad p_4 = -\frac{2\theta_a}{T_g^3}$ $q_1 = -4\theta_a \ \ q_2 = 12\frac{\theta_a}{T_g}$ $q_3 = -\frac{9\theta_a}{T_g^2} \ \ q_4 = 2\frac{\theta_a}{T_g^3}$
	$\theta_{4T}(t_1) = \pi,\ \theta_{4T}(t_2) = \pi - \theta_b,$ $\dot{\theta}_{4T}(t_1) = 0,\ \dot{\theta}_{4T}(t_2) = 0,$	$\theta_{4T}(t) = \pi\ t \in (t_0, t_1)$ $\theta_{4T}(t) = r_1 + r_2t + r_3t^2 + r_4t^3$ $t \in (t_1, t_2)$	$r_1 = \pi - 5\theta_b \quad r_2 = \frac{12\theta_b}{T_g}$ $r_3 = \frac{-9\theta_b}{T_g^2}, \quad r_4 = \frac{2\theta_b}{T_g^3}$
	Phase-2 $(t \in (t_2, t_4))$		
	$x_{A_2}(t_2) = x_{A_1}(t_2) = x_e, \dot{x}_{A_2}(t_4) = 0,$ $\dot{x}_{A_1}(t_2) = \dot{x}_{A_2}(t_2) = x_v, x_{A_2}(t_4) = x_f.$	$x_{A_2}(t) = n_0 + n_1t + n_2t^2 + n_3t^3$	$N_{4\times 1} = A^{-1}_{4\times 4}.M_{4\times 1}$
	$z_{A_2}(x_e) = z_{A_1}(t_2) = z_e, z_{A_2}(x_f) = z_f,$ $\dot{z}_{A_2}(x_e) = \dot{z}_{A_1}(t_2) = z_v,$ $\dot{z}_{A_2}(x_m) = h_1,\ x_m = \frac{x_f}{2},$	$z_{A_2}(t) = a + bx_{A_2}(t) + cx_{A_2}(t)^2$ $+dx_{A_2}(t)^3$	$C_{4\times 1} = B^{-1}_{4\times 4}.X_{4\times 1}$
	$x_{A_2}(t_2) = x_{A_1}(t_2) = x_e, x_{A_2}(t_4) = x_f,$ $\dot{x}_{A_2}(t_2) = \dot{x}_{A_1}(t_2) = x_v, \dot{x}_{A_2}(t_4) = 0,$ $\ddot{x}_{A_2}(t_2) = a_c.$	$x_{A_2}(t) = n_0 + n_1t + n_2t^2$ $+n_3t^3 + n_4t^4$	$N_{5\times 1} = A^{-1}_{5\times 5}.M_{5\times 1}$
	$z_{A_2}(x_f) = z_f, z_{A_2}(x_e) = z_{A_1}(t_2) = z_e,$ $\dot{z}_{A_2}(t_2) = \dot{z}_{A_1}(x_e) = z_v, z_{A_2}(x_m) = h_1,$ $\dot{z}_{A_2}(x_m) = 0.$	$z_{A_2}(t) = a + bx_{A_2}(t) + cx_{A_2}(t)^2$ $+dx_{A_2}(t)^3 + ex_{A_2}(t)^4$	$C_{5\times 1} = B^{-1}_{5\times 5}.X_{5\times 1}$
	$x_{A_2}(t_2) = x_{A_1}(t_2) = x_e, x_{A_2}(t_4) = x_f,$ $\dot{x}_{A_2}(t_2) = \dot{x}_{A_1}(t_2) = x_v, \dot{x}_{A_2}(t_4) = 0,$ $\ddot{x}_{A_2}(t_2) = a_c, \ddot{x}_{A_2}(t_4) = a_{c1}$	$x_{A_2}(t) = n_0 + n_1t + n_2t^2 + n_3t^3$ $+n_4t^4 + n_5t^5$	$N_{6\times 1} = A^{-1}_{6\times 6}.M_{6\times 1}$

Table 4.2: For trajectories of swing leg

During both the phases, the swing leg works like a 2-link (HK and KA) manipulator treating hip H as a moving base and ankle joint A as end effector. The foot of the swing leg moves along with the ankle joint in SSP (See Figure 4.4(a) and 4.4(b)). Hence, the forward kinematic equations of the swing leg are

$$l_1 cos\theta_1(t) + l_2 cos(\theta_1(t) + \theta_2(t)) = x_A(t) - x_H(t); \tag{4.15}$$

$$l_1 sin\theta_1(t) + l_2 sin(\theta_1(t) + \theta_2(t)) = z_A(t) - z_H(t); \tag{4.16}$$

where $x_A(t), z_A(t)$ and $x_H(t), z_H(t)$ are x and z trajectories of the ankle and hip respec-

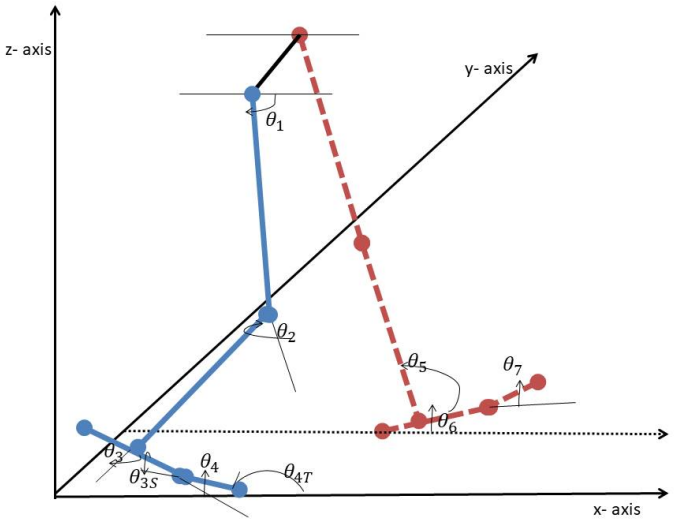


Figure 4.5: Both legs with joint angles

tively, are modeled in both phases as given in Tables 4.2 and 4.3, so the RHS of Equations 4.15 and 4.16 are known at each time instant t. Hence $\theta_1(t)$ and $\theta_2(t)$ can be obtained by solving the inverse kinematics using FNN as discussed in Section 3.4.1.

4.3.2 Stable Leg's Joint Trajectories

Assume that the stable leg moves on the line $y = l_0$. Biped do not fold the stable leg During both phases of walk and most of the time in SSP the whole body weight is on the stable leg similar to human walk.

As in Figure 4.3(c), the links EsSs and SsTs respectively, represents sole and toe of the stable leg's foot during DSP. The two phases of the stable foot's trajectory are discussed below.

Phase-1: During DSP $(0, t_2)$, stable leg's heel joint(Es) is stationary at $(x_{f1}/2, l_0, 0)$ while $EsTs$ moves till it coincides with the line $y = l_0$(Figure 4.3(c)). It is assumed that the foot EsTs starts with an angle θ_a at t=0 and comes in contact with the line$y = l_0$ at $t = t_2$. During time interval $[0, t_1]$, sole joint moves in a circular path with center at Es and radius $2l_3$. Simultaneously the toe joint also moves in a circular path with center S and

radius l_4.

In time interval $[t_1, t_2]$, the toe SsTs reverses its circular motion with base at S while the sole EsSs is flat on the line $y = l_0$. Joint angles θ_6 and θ_7 are given in the Table 4.3. Designed parameters θ_a and θ_b are the maximum angles subtended by EsSs and SsTs respectively with the line $y = l_0$.

The forward kinematic equations for the stable leg's foot as 2-link (EsSs and SsTs) manipulator with Es as the base and Ts as the end effector are given by

$$2l_3 cos\theta_6(t) + l_4 cos(\theta_6(t) + \theta_7(t)) = x_{Ts}(t) - \frac{x_{f1}}{2} \tag{4.17}$$

$$2l_3 sin\theta_6(t) + l_4 sin(\theta_6(t) + \theta_7(t)) = z_{Ts}(t) \tag{4.18}$$

x_{Ts}, z_{Ts} are the x and z coordinates of the stable foot's toe joint.

Phase(2) : During SSP, stable foot is stationary on the ground. Both the hips (H and

	Trajectories for Stable leg		
	Boundary conditions	**Trajectory**	**Coefficients**
	Toe joint angles		
	$\theta_7(t_0) = 0, \theta_7(t_1) = \theta_b,$ $\dot{\theta}_7(t_0) = 0, \ \dot{\theta}_7(t_1) = 0,$ $\theta_7(t_1) = \theta_b, \ \theta_7(t_2) = 0,$ $\dot{\theta}_7(t_1) = 0, \ \dot{\theta}_7(t_2) = 0,$	$\theta_7(t) = p_1 + p_2 t + p_3 t^2 + p_4 t^3$ $t \in (t_0, t_1)$ $\theta_7(t) = q_1 + q_2 t + q_3 t^2 + q_4 t^3$ $t \in (t_1, t_2)$	$p_1 = 0 \quad p_2 = 0$ $p_3 = \frac{3\theta_b}{T_g^2} \quad p_4 = -\frac{2\theta_b}{T_g^3}$ $q_1 = -4\theta_a \ \ q_2 = 12\frac{\theta_a}{T_g}$ $q_3 = -\frac{9\theta_a}{T_g^2} \ \ q_4 = 2\frac{\theta_a}{T_g^3}$
	Sole joint angles		
	$\theta_6(t_0) = \theta_a, \ \theta_6(t_1) = 0,$ $\dot{\theta}_6(t_0) = 0, \ \dot{\theta}_6(t_1) = 0,$	$\theta_6(t) = r_1 + r_2 t + r_3 t^2 + r_4 t^3$ $t \in (t_0, t_1)$ $\theta_6(t) = 0 \quad t \in (t_1, t_2)$	$r_1 = \theta_a \quad r_2 = 0$ $r_3 = \frac{-3\theta_a}{T_g^2}, \quad r_4 = \frac{2\theta_a}{T_g^3}$
	Hip trajectory		
	$x_H(t_0) = l_3 + x_f/4, x_H(t_4) = l_3 + 3x_f/4,$ $\dot{x_H}(t_0) = v_h, \dot{x_H}(t_4) = v_h,$	$x_H(t) = r_1 + r_2 t + r_3 t^2 + r_4 t^3$ $z_H(t) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_{As}(t))^2};$	$r_1 = l_3 + \frac{x_f}{4}, \ r_4 = -2(\frac{x_f}{2t_4^3} - \frac{(v_h + v_h)}{2t_4^2}),$ $r_2 = v_h, \ r_3 = -r_4 \frac{3t_4}{2}.$

Table 4.3: For trajectories of stable leg

Hs) are following the same trajectory in sagittal plane. The joint H follows a circular path considering the joint As of stable leg as center and $(l_1 + l_2)$ as radius (See Figure 4.4) with suitable boundary conditions,. At $t = 0$, the coordinate of stable leg's hip joint H is $(l_3 + \frac{x_f}{4}, l_0, h)$ where h is the initial hip height. During the time interval $(0, t_4)$, the x and z trajectories of joint H are given in Table 4.3.

Stable leg HA works like a 1-link manipulator as there is no rotation at knee joint.

Hence, the forward kinematic equations of the stable leg are

$$(l_1 + l_2)cos\theta_5(t) = x_H(t) - x_{As} \tag{4.19}$$

$$(l_1 + l_2)sin\theta_5(t) = z_H(t) - z_{As} \tag{4.20}$$

where $x_{As}(t) = \frac{x_{f1}}{2} + l_3 \cos\theta_6(t)$ and $z_{As}(t) = l_3 \sin\theta_6(t)$; respectively, are the x and z coordinates of the stable leg's ankle joint. The inverse kinematic solutions for these trajectories can be obtained by using FNN (see Section 3.4.1).

4.4 ZMP Stability Analysis and Upper Body Motion

4.4.1 ZMP Stability

ZMP is defined in Section 2.6.3 of Chapter 2. In Figure 4.6, supported region for SSP

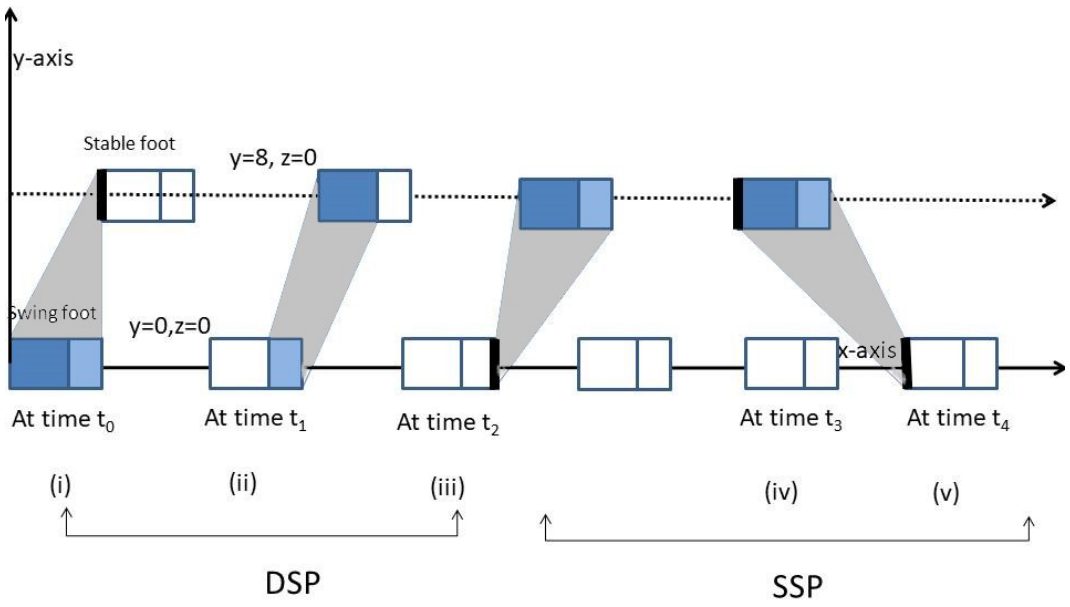


Figure 4.6: Supported region for SSP and DSP during one cycle(step)

and DSP at some time instant are given. Light blue region represents the toe and the dark blue region represents the sole of the foot. Blue shaded region shows that the foot is on the ground while white region shows that the foot is above the ground.

Gray shaded region between two feet shows convex hull of supporting points. Due to the toe and the sole movements, stability region is not fixed and hence it changes at every instant of time(Figure 4.6).

In DSP $(0, t_2)$, the flexible swing foot first lifts the heal(E) from the ground at $t = 0$ (Figure 4.6(i)), then lifts the sole joint S at t=t_1 (Figure 4.6(ii)) and finally lifts the toe tip T at t=t_2 (Figure 4.6(iii)) to leave the ground. Simultaneously, the stable foot whose heal is on the ground and the toe above the ground moves toward the ground to coincide with the line $y = l_0$. During the motion of both the feet, the upper body mass is shifts from the swing leg's side to the stable leg's side. Because of this, the ZMP moves from swing foot's toe position to a region below the stable foot (Figure 4.6(ii)). In SSP (t_2, t_4), due to the swing leg's forward motion, the ZMP moves continuously from heel to toe of the stable foot.

4.4.2 Upper Body Motion

Modern walking robots usually have heavy upper body as electronic circuits and batteries are there. Thus, the upper body mass has its effect on the stability. Therefore, the parameters for the upper body motion are changed in given ranges to find the ZMP trajectory which moves in a desired manner. If the upper body is in the upright position then, it reduces the risk of falling down and a similar phenomena is observed in the human gait also. The total mass of the upper body is assumed to be at single point(U) for planning its trajectory.

Upper Body Motion on Frontal Plane

On the frontal plane (yz plane), upper body mass(U) shifts from one side to another. Here, three cases of U trajectory in y-direction are considered. These trajectories are determined by cubic polynomials.

Case-1: As robot takes its step, U starts to move from swing leg's hip to stable leg's hip during DSP (t_0, t_2) and stays there all the time during SSP (t_2, t_4). Then, the U trajectory in y-direction is given by:

$$y_U(t) = \begin{cases} y_0 + y_v t + \frac{-2y_v t_2 + 3(y_a + y_m/2 - y_0)}{t_2^2} t^2 + \frac{-2(y_a + y_m/2 - y_0) + y_v t_2}{t_2^3} t^3 & t_0 \leq t \leq t_2 \\ y_a & t_2 \leq t \leq t_4 \end{cases}$$

Case-2: As the robot takes its step, U starts to move from the middle of both hips to

the side of the stable leg's hip in the time interval (t_0, t_1), stays there during time t_1 to t_3 and again starts moving towards the middle of both hips between in the time interval (t_3, t_4). Then, the U trajectory in y-direction is given by:

$$y_U(t) = \begin{cases} \frac{y_l}{2} + \frac{3(y_l+y_a)}{(4{t_1}^2)}t^2 - \frac{y_l+y_a}{2({t_1}^3)}t^3 & t_0 \le t \le t_1 \\ y_a & t_1 \le t \le t_3 \\ (\frac{(16(t_3-t_4)^3-(9(y_l+y_a)t_4^2t_3-(y_l+y_a)t_4^3)}{4(t_3-t_4)^3} & \\ +(\frac{3(y_l+y_a)}{2(t_3-t_4)^3}t_4t_3)t + \frac{3(y_l+y_a)(t_4+t_3)}{4(t_3-t_4)^3}t^2 - \frac{(y_l+y_a)}{2(t_3-t_4)^3}t^3 & t_3 \le t \le t_4 \end{cases}$$

Case-3: As the robot takes its step, U starts to move from the middle of both hips to the side of the stable leg's hip in the time interval (t_0, t_2) and again starts moving towards the middle of both hips between t_2 to t_4 in y-direction.
Then, the U trajectory in y-direction is given by:

$$y_U(t) = \begin{cases} \frac{y_l}{2} + \frac{3(y_l+y_a)}{(4{t_2}^2)}t^2 - \frac{y_l+y_a}{2({t_2}^3)}t^3 & t_0 \le t \le t_2 \\ (\frac{(16(t_2-t_4)^3-(9(y_l+y_a)t_4^2t_2-(y_l+y_a)t_4^3)}{4(t_2-t_4)^3} + & \\ (\frac{3(y_l+y_a)}{2(t_2-t_4)^3}t_4t_2)t + \frac{3(y_l+y_a)(t_4+t_2)}{4(t_2-t_4)^3}t^2 - \frac{(y_l+y_a)}{2(t_2-t_4)^3}t^3 & t_2 \le t \le t_4 \end{cases}$$

Here, y_0 is initial position of U, $y_l = l_0/2$ is the middle position of both hips, y_a is final position of U and v_{y_0} is initial velocity.

4.5 Simulation Results

Initially, swing foot lies in the rectangular region, $0 \le x \le 6$, $-1.5 \le y \le 1.5$ and stable foot lies in the region given by rectangle $6.86 \le x \le 12.86$ and $6.5 \le y \le 9.5$. Maximum sole rotation $\theta_a = \pi/6$ rad, maximum toe rotation $\theta_b = \pi/6$ rad are the designed parameters. The values of these parameters are chosen to resemble a normal human of height 40 units. Simulation are done with three types of ankle trajectory(see Table 4.2) along with three cases of upper body motions (as given in Section 4.4.2) to get most suitable ZMP trajectory. In the following, this approach is discussed briefly.

4.5.1 Robot Leg Trajectories

Figures 4.7, 4.8 and 4.9 show x and z positions, velocities and accelerations graphs for the ankle trajectories which are provided in Table 4.2. Parameters for these trajectories are: x_f=14 units, t_f=2.1 secs and h_1=2.5 units.

Inverse kinematic solutions of the equations 4.15-4.16 and 4.17-4.18 are calculated using FNN for given related positions of feet and hips. In Figures 4.10(a), (b) and (c), the swing leg follows cubic, quartic and quintic ankle trajectories respectively and in Figure 4.10(d) stable leg follows the hip trajectory. Figure 4.11(a) and (b) shows the swing leg's foot and stable leg's foot movements respectively in DSP$(0, t_2)$.

4.5.2 Effect of Upper Body Motion on ZMP Stability

While walking slowly biped robot can be stable without upper body but during fast walking there should be a upper body. To verify this x-ZMP and y-ZMP trajectories (i) without upper body, (ii) with fixed upper body (at the middle of the hip) and (iii) with upper body motion are compared in Figure 4.12. By observing the Figures 4.12(a) and (b), it can be concluded that there should be upper body motion in y-direction to get suitable x-ZMP and y-ZMP trajectories but the particular manner in which the upper body should moves is unknown. For this purpose, three cases of upper body trajectories are considered as in Section 4.4 (see Figure 4.13).

From Figure 4.14(a), (b) and (c), it is observed that the best upper body trajectory for ZMP stability is obtained using Case-1 (Figure 4.14(a)), in which upper body shifts from swing leg's hip to stable leg's hip during DSP and fixed there during SSP.

4.5.3 Effect of Ankle Trajectory on ZMP Stability

Ankle trajectory is already discussed in subsection 4.3.1. Now, we observe its effect on stability.

As in Figure 4.9(a), x-acceleration for cubic polynomial is discontinuous at time $t = t_2$. It affects the ZMP stability as in Figure 4.15(a), 4.16(a) and 4.17(a). To avoid this, x and z trajectories are generated by quartic polynomial and we get continuous x and z acceleration trajectories for ankle joint as in Figure 4.9(b). However, x-acceleration at time $t = t_4$ is very large (Figure 4.9(b)) and it may also affect the ZMP stability as in Figure 4.15(b), 4.16(b) and 4.17(b). To avoid this, final acceleration is assumed to be 0 and the x trajectory

of ankle is generated by quintic polynomial using 6 boundary conditions (Figure 4.9(c)). Now, we get the suitable ZMP trajectory as in Figure 4.15(c), 4.16(c) and 4.17(c). Details observations are given in Table 4.4.

4.5.4 Selection of Parameters for Stable ZMP Trajectory

There are several parameters which highly affect the ZMP stability such as hip velocity, step time, step length, step height, and initial velocity of upper body etc. Now, the values of parameters are changed within the reasonable range to find ZMP trajectory with largest stability margin. A large number of simulations is done with different set of walking parameters. The parameters are varying as x_f from 12 to 16 units, time t_f from 2 to 3 secs, velocity of hip from 2.5 to 3.5 units/sec, and step height h_1 from 2.5 to 4 units for cubic, quartic and quintic polynomial trajectories. The best set of parameters for these trajectories are analyzed whichever gives most stable ZMP trajectory are highlighted in Table 4.4.

Final ZMP Trajectory for 3 Steps

Figures 4.18(a), (b), (c) and (d) demonstrate the final x and z COM position and velocity graphs for each joint and Figure 4.18(e) demonstrates final COM trajectory of whole body. The final ZMP graph for joint trajectories with optimum parameters (as highlighted in Table 4.4) is given in Figure 4.18(f). In the Figure 4.18(f), the ZMP varying trajectory is inside the support polygon that provides a stable walking of the biped robot for 3 steps.

Full body motion of the biped robot in 3 dimensional space for 2 steps is given in Figure 4.19.

4.5.5 Robot Walk on Uneven Surface

The Obstacle Crossing

As in Figure 4.20, the biped robot can cross an obstacle of width 3.5 units and height 5.5 units using a step length 16 units with step height 9 units within 4 seconds. An increment of 3 units/sec in x velocity and 2 units/sec in z velocity of ankle at time t_2(time at which DSP change into SSP) is taken.

Upper body	Ankle trajectory	Step Time	Initial hip velocity(in/sec)	upper body parameters	ZMP stability	Conclusion
No upper body	cubic,quartic,quantic	4.5	v_h=2.95	-	-	x-ZMP is in the region for $t > 4.5s$ but y-ZMP at middle of hip
Fixed	cubic,quartic,quantic	3	v_h=2.55	$y_l = \frac{l_0}{2}$	-	x-ZMP is in the region for $t > 3s$ but y-ZMP at middle of hip
Moving						
Case-2	cubic,quartic,quantic	3	v_h=2	$y_v = 0$,$y_l = \frac{l_0}{2}$,$y_a = 8$	unstable	as for last 4 time instant ZMP is out of region
Case-3	cubic,quartic,quantic	3	v_h=2	$y_v = 0$,$y_l = \frac{l_0}{2}$,$y_a = 8$	unstable	as for last 4 time instant ZMP is out of region
Case-1	cubic	2.5	v_h=2.8	$y_v = 7$,$y_0 = 0$,$y_a = 8$	stable	stable with desired stability margin
	cubic	2.2	v_h=3.15	$y_v = 8$,$y_0 = -.5$,$y_a = 8.5$	unstable	
	quartic	2.5	v_h=2.8	$y_v = 8$,$y_0 = 0$,$y_a = 8$	stable	stable with desired stability margin
	quartic	2.2	v_h=3.15	$y_v = 9$,$y_0 = -.5$,$y_a = 8.5$	stable	stable with small stability margin
	quartic	2.1	v_h=3.55	$y_v = 9$,$y_0 = -.5$,$y_a = 8.5$	unstable	as for last 2 time instant ZMP is out of region
	quintic	2.5	v_h=2.8	$y_v = 8$,$y_0 = 0$,$y_a = 8$	stable	stable with desired stability margin
	quintic	2.2	v_h=3.15	$y_v = 9$,$y_0 = 0$,$y_a = 8$	stable	stable with desired stability margin
	quintic	**2.1**	v_h=3.55	$y_v = 9$,$y_0 = -.5$,$y_a = 8.5$	**stable**	**stable with desired stability margin**
	quintic	2	v_h=3.65	$y_v = 9$,$y_0 = -.5$,$y_a = 8.5$	stable	stable with small stability margin
	quintic	1.8	v_h=3.65	$y_v = 9$,$y_0 = -.5$,$y_a = 8.5$	unstable	

Table 4.4: Simulations based observations

Walking on Irregular Surface

The robot can also walk on irregular surface of height 0 to 1 unit only by adjusting the foot (ankle plus toe) movements. Here, the biped motion is presented for 2 different cases of walk on irregular surface: (i) foot at a height and (ii) foot at inclined upward/downward position(see Figure 4.21).

Let the robot walks on irregular surface for three steps. In first step, let swing foot lands flatly on irregular surface of height 1 unit. ZMP trajectory of this case is given in Figure 4.23(a) which is approximately same as the plane surface trajectory.

In second step, let swing foot lands in a inclined upward position with toe joint on irregular surface of height$< 1/2$ unit (so only toe joint angle is changed see Figure 4.22(b)) while stable foot now become flat on the obstacle because of the previous step. So the base for hip trajectory is slightly changed due to the position of the stable foot. In this case, ZMP trajectory is slightly changed as in Figure 4.23(b).
Now in third step, let stable foot is at inclined upward position with toe joint on irregular surface while swing foot lands at inclined upward position with sole joint on irregular surface of height$< 1/2$ unit, so results are similar to second step.

4.5.6 Comparison with Existing Models

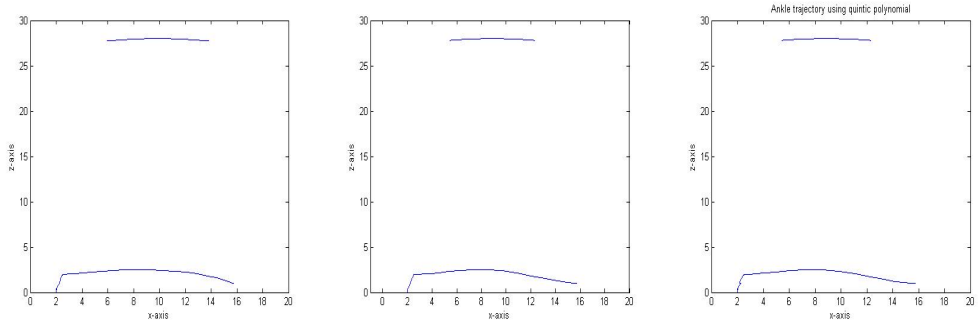
The comparison of proposed model with existing models is given in Table 4.5.

Ref.No.	Foot			Upper body movements	Stable leg's Knee bend all time	speed	Surface	ZMP Stability
	(length, width)	Toe and Sole Rotation	Gap in between feet (x-direction, y-direction)					
[14]	(10.4, 6.6)	No	(0, 3.4)	Upright move only in y-direction	No	5units/s	plane	Stable with less stability margin
[10]	(10, 8)	No	(0, 4)	all time bend in x-direction	Yes	5units/s	plane and slightly uneven	Good Stable on plane surface but less stable on uneven surface
Proposed	(6, 3)	Yes	(1, 5)	Upright move only in y-direction (frontal plane)	No	6.7units/s	plane and slightly uneven	Good stable on both plane and uneven surface and can cross obstacles

Table 4.5: Comparison table

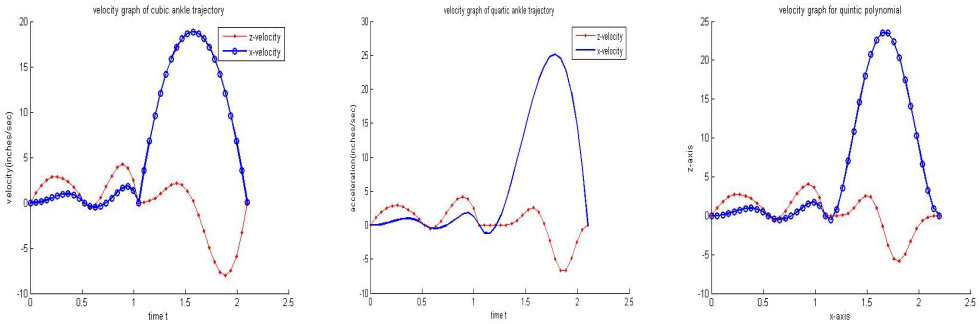
4.6 Conclusion

In this work, a biped model is proposed which has human like gait with ZMP stability without bending knee and upright upper body while walking on plane and uneven surfaces. In the proposed walking patterns, the robot can walk at low and medium speeds. Here, first the joint trajectories are generated and then the walking stability of robot is ensured by ZMP stability criterion. Movements at ankle and toe joint are added for realizing natural human like walk.



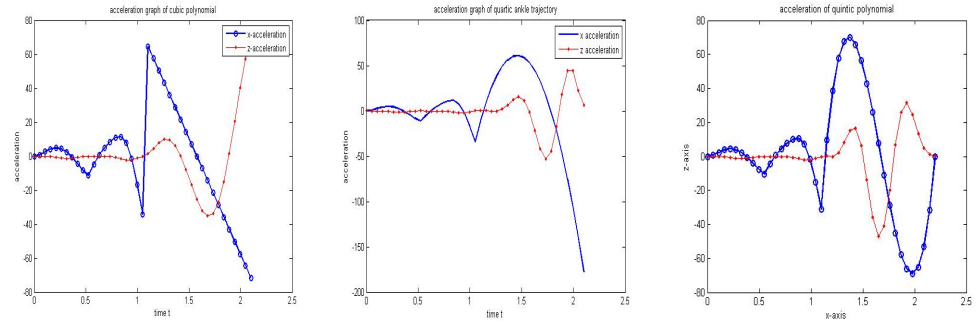
(a) Cubic ankle trajectory (b) Quartic ankle trajectory (c) Quantic ankle trajectory

Figure 4.7: Ankle and hip trajectories for $t\epsilon(0, t_4)$



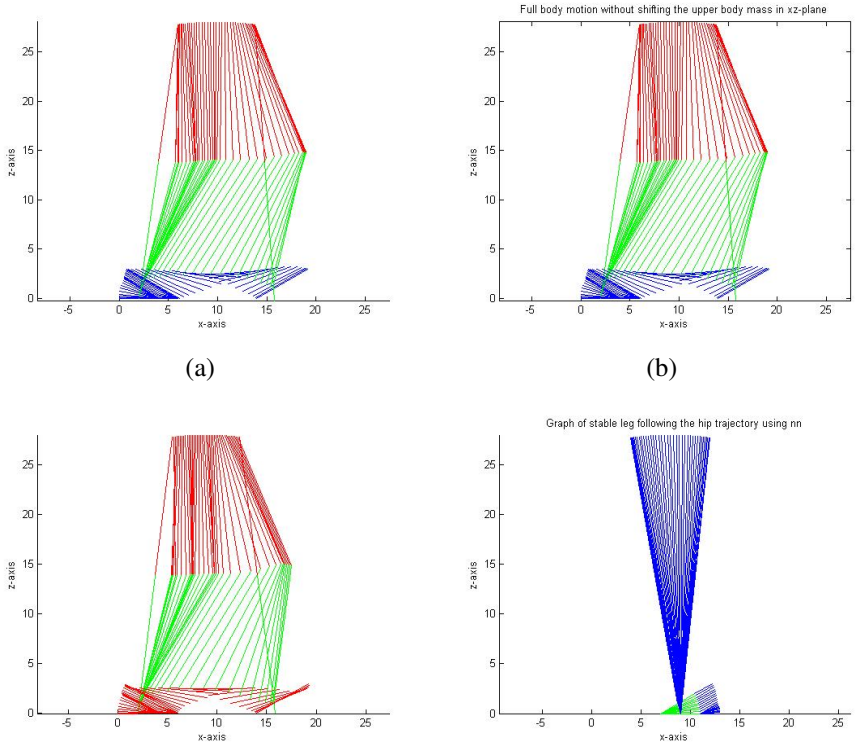
(a) Cubic (b) Quartic (c) Quantic

Figure 4.8: Ankle velocity trajectory for $t\epsilon(0, t_4)$



(a) Cubic (b) Quartic (c) Quantic

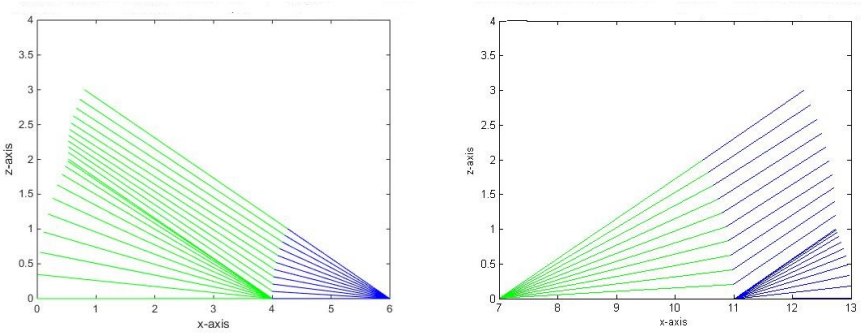
Figure 4.9: Ankle acceleration trajectory for $t\epsilon(0, t_4)$



(a) (b)

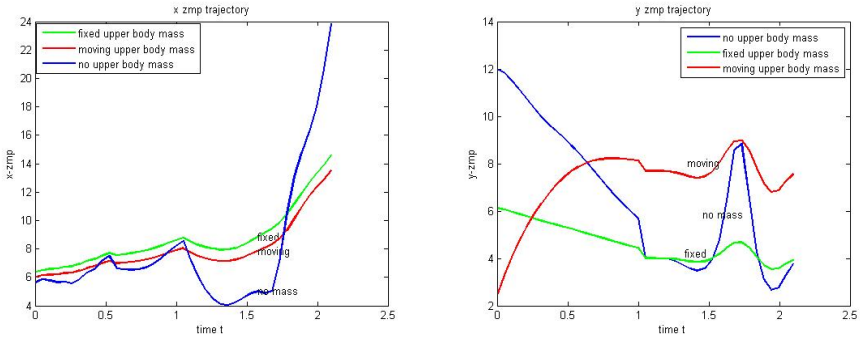
(c) (d)

Figure 4.10: Legs following the given trajectories using FNN for $t\epsilon(0, t_4)$



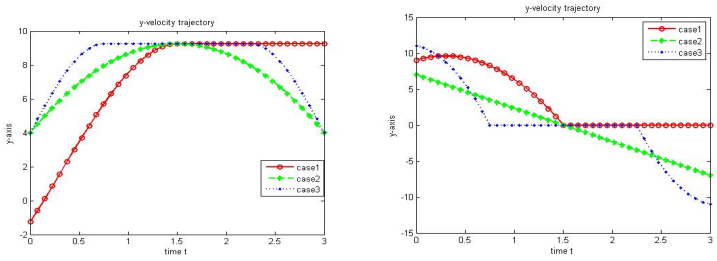
(a) (b)

Figure 4.11: Feet movements during DSP $t\epsilon(0, t_2)$



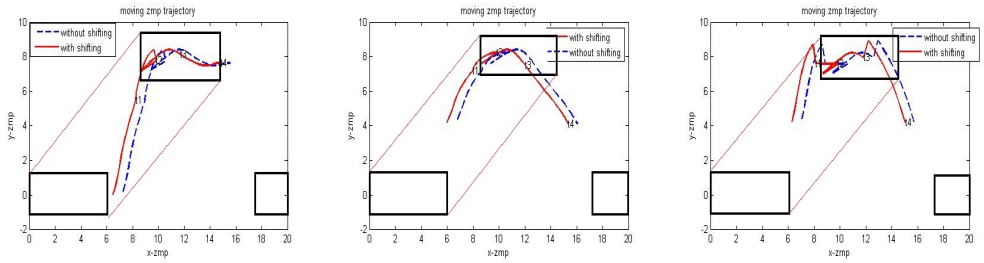
(a) x-ZMP (b) y-ZMP

Figure 4.12: Effect of upper body on ZMP



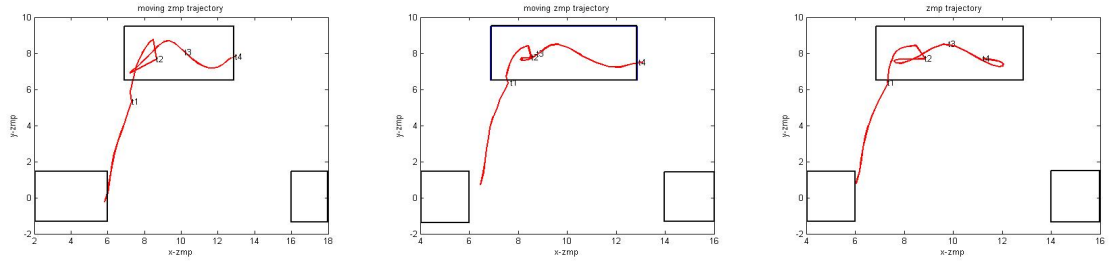
(a) (b)

Figure 4.13: Position and velocity graph for 3 different cases of upper body motion



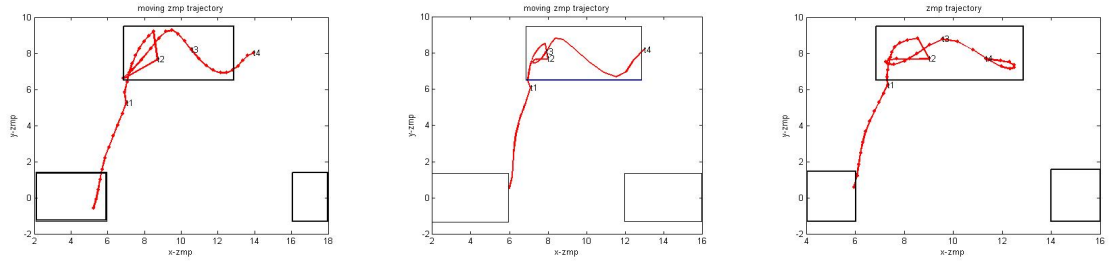
(a) Case-1 (b) Case-2 (c) Case-3

Figure 4.14: 3 different cases of upper body motion



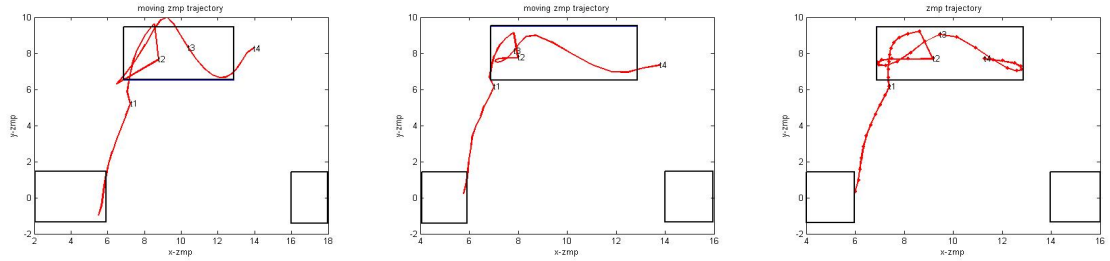
(a) (b) (c)

Figure 4.15: ZMP trajectory in 2.5 seconds for 3 types of ankle trajectories



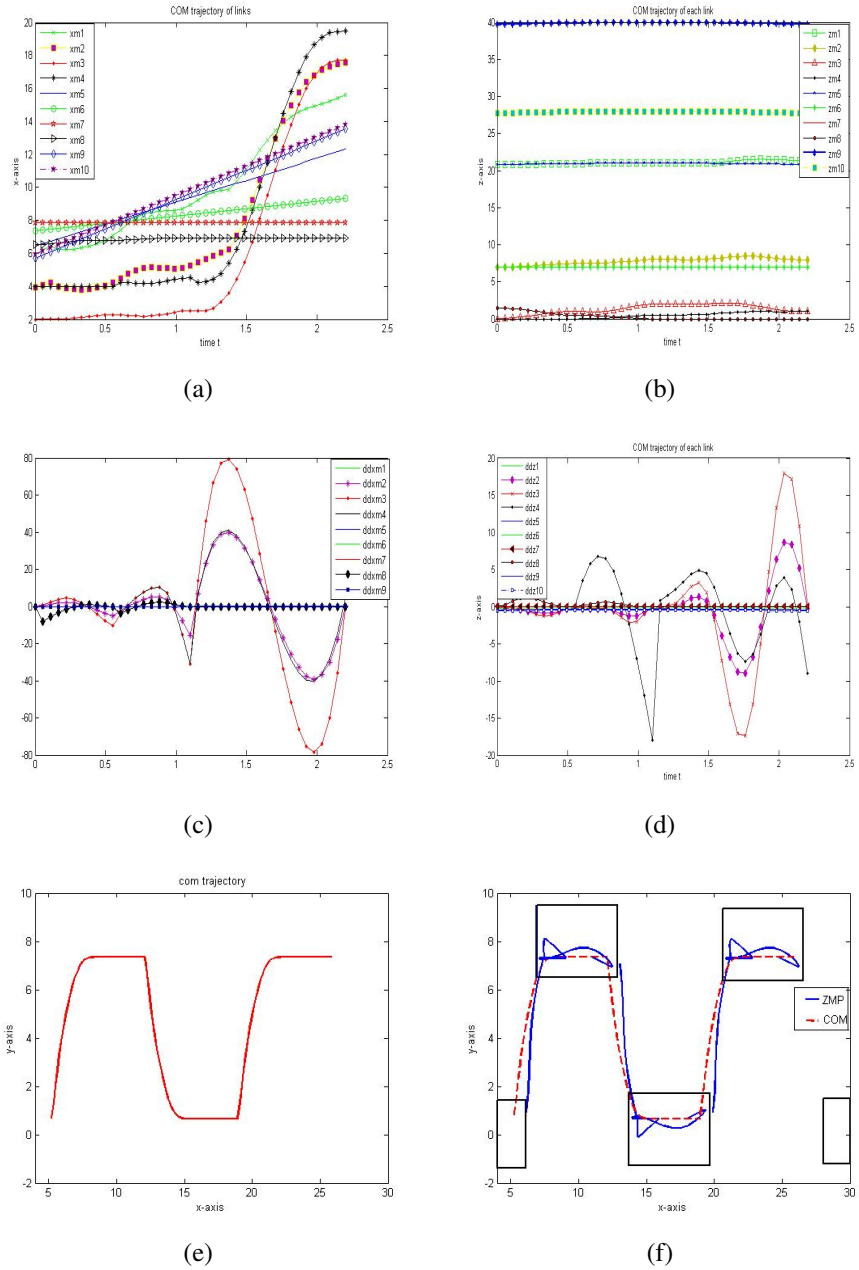
(a) (b) (c)

Figure 4.16: ZMP trajectory in 2.2 seconds for 3 types of ankle trajectories



(a) (b) (c)

Figure 4.17: ZMP trajectory in 2 seconds for 3 types of ankle trajectories



(a) (b)

(c) (d)

(e) (f)

Figure 4.18: For quintic polynomial in 2.1 sec

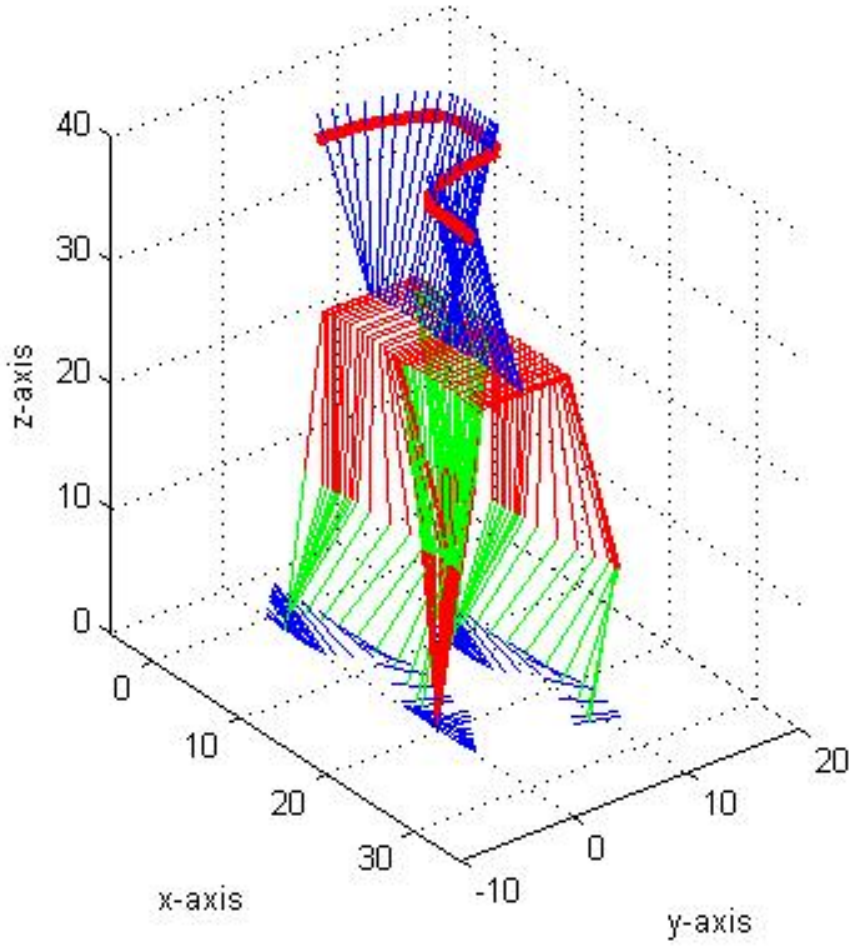
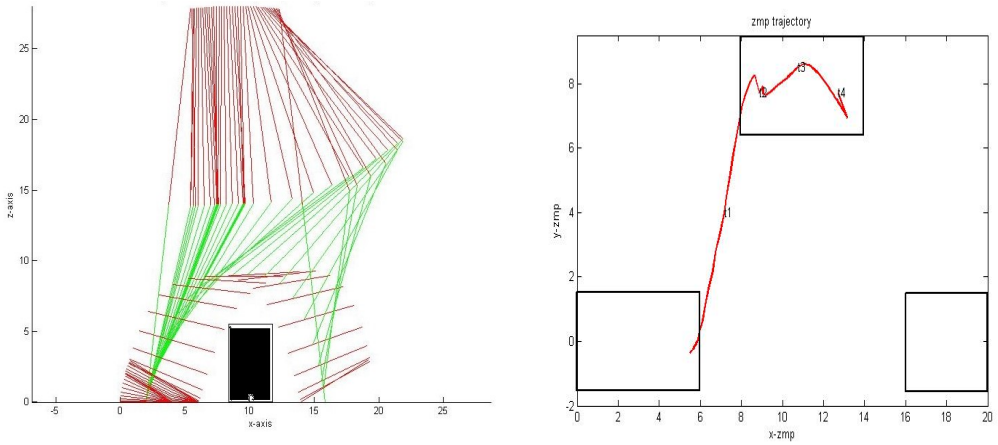


Figure 4.19: Full body motion in 3D for 2 step



(a) (b)

Figure 4.20: Crossing an obstacles

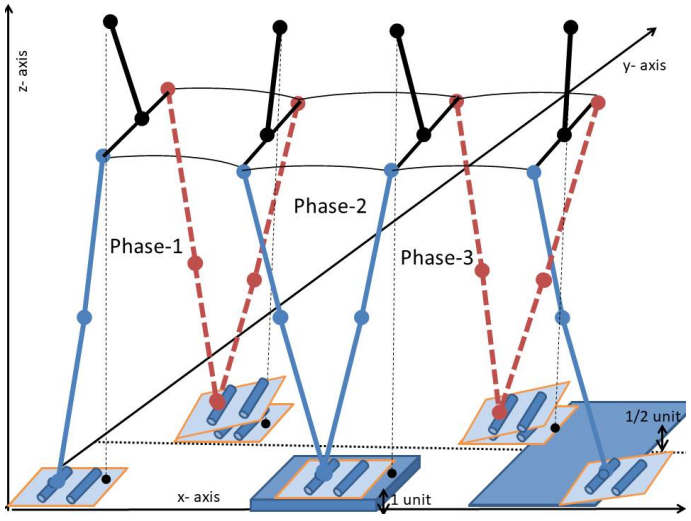
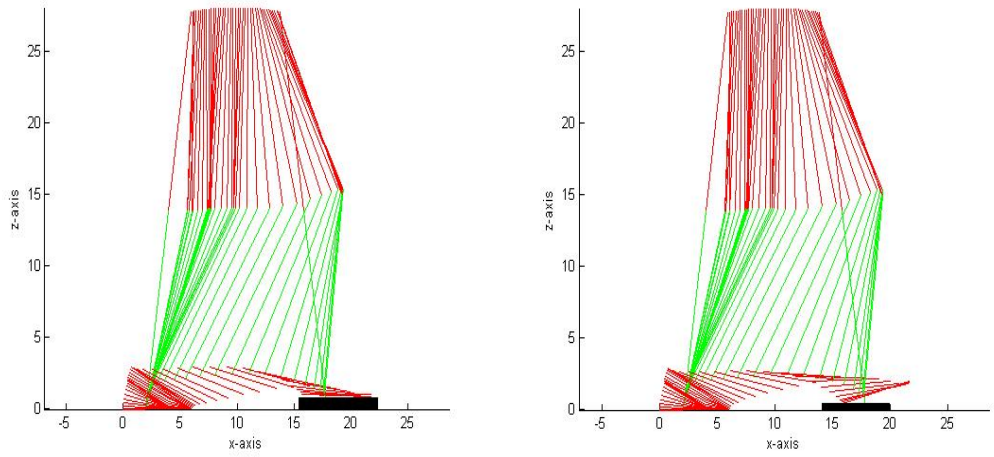
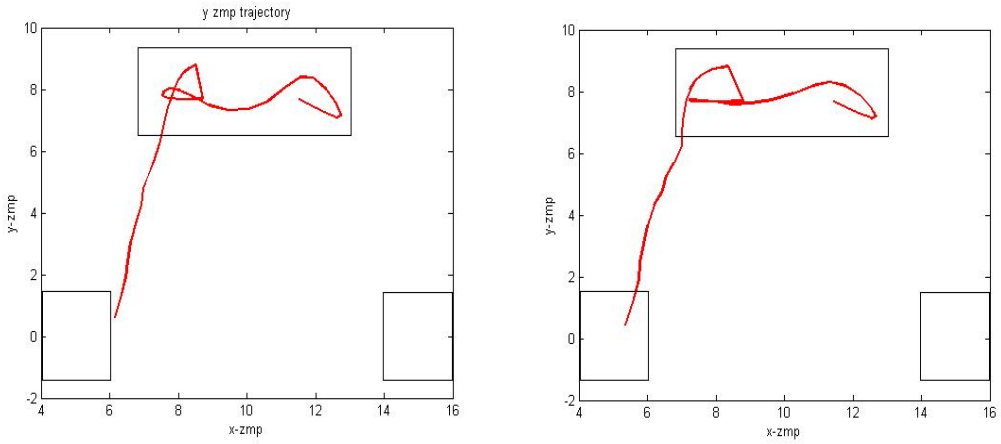


Figure 4.21: 3 steps on irregular surface



(a) (b)

Figure 4.22: Different cases of irregular surface



(a) (b)

Figure 4.23: ZMP trajectory for Figure 24

Chapter 5: Stable Gait Generation for flat footed Biped Robot using FNN and WNN

Feedforward Neural Network (FNN) and Wavelet Neural Network (WNN) based trajectory generation approaches are proposed for smooth and dynamically stable walk of a 5 DOF biped robot on flat terrain in 3-dimensional space. The trajectory generation procedure is derived using semi-supervised NN for given boundary conditions without assigning any path in advance. The suitability of the proposed approaches is studied using Zero Moment Point (ZMP) stability criteria and simulations have been carried out using Matlab2014a.

5.1 Introduction

The biped robot should be able to walk according to different surface conditions and according to application in different fields. So, stable biped locomotion is still a challenging problem in robotics. Trajectory generation problems have been extensively studied in the literature [15, 22, 31, 66, 67, 100]. However, the predesigned trajectories failed whenever the terrain conditions change during walk. Humans do not walk based on a predefined trajectory rather they modify the trajectory during tracking

Recently different soft computing techniques namely, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Fuzzy Logic, Neural Network (NN) have been investigated for motion planning of robots [12, 43, 55, 59, 60, 82]. Several authors have proposed NN models to resolve the trajectory planning problems [1, 30, 56, 60, 77, 97]. Vundavilli et al. [97] used NN based gait on a sloping surface. In Semwal et al. [84], a multilayered back propagation neural network is used for gait pattern classification. Capi et al. [13] used RBFNN to optimize the pre-computed gait for minimum energy consumption. In Dutta et al. [26], FNN based optimization procedure is used to calculate the optimum coefficient of

polynomial for given constraints. In some papers, trajectories are generated by NN using sensory feedback of data [56, 105], or by collecting human data [84]. All these papers used NN either for parameter optimization or for data optimization and some paper used NN controller to control the motion. NN approach can be effective for generating trajectory since it gives instant result at desired points in negligible computation time and memory without reconstruction, once the networks have been trained.

In this chapter, novel FNN and WNN based trajectory generation procedures are presented to find the smooth trajectories in less computing time with the help of various sets of boundary conditions/constraints at various time points. First, a trial trajectory is defined as a function which is the sum of two networks in such a way that first network vanishes at initial point and second network vanishes at final point. Then, both the networks are trained to satisfy the boundary conditions by minimizing the corresponding error. Now, the trained function is the required trajectory for the given boundary conditions. In case more constraints need to be imposed on the trajectory, more neural network function can be added and trained accordingly. FNN trajectories are generated for hip and ankle joints and for upper body to walk on a plane surface. ZMP stability criteria is used to ensure the dynamic balance for these NN trajectories.

The organization of this chapter is as follows. Robot model is described in Section 5.2. In Section 5.3, methodology for trajectory generation based on FNN and WNN is developed and the simulated results are compared. In Section 5.4, NN based trajectory generation procedure for feet, hip and upper body are discussed and simulated results are also given. Conclusion is given in Section 5.5.

5.2 Modelling of Biped Robot

5.2.1 Robot Design

As in Figure 5.1, the biped robot has 2 Degrees of Freedom (DOF) for each leg with flat foot and one DOF for Upper body(U). Hip joint, Knee joint, Ankle joint of swing leg and stable leg are represented by H, K, A and Hs, Ks, As respectively. All the joints are revolute joints. The parameters of the robot are given in Table 5.1. Total length of a leg is $(l_1 + l_2)$. It is assumed that the length and mass of both legs are the same.

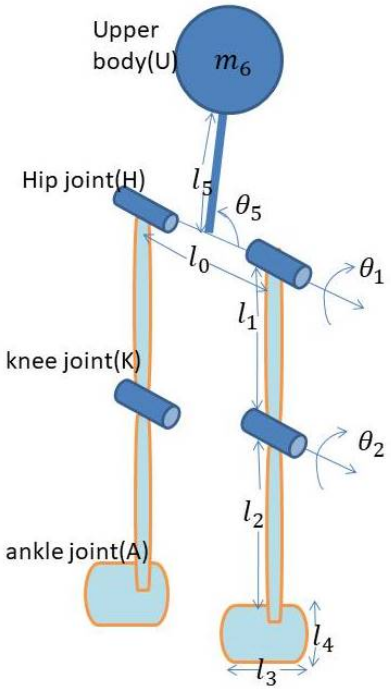


Figure 5.1: Robot Structure

Link	Length	value
HK	l_1	14 units
KA	l_2	14 units
foot length	l_3	6 units
HU	l_5	12 units
HH	l_0	8 units
foot width	l_4	3 units
Link	Mass	value
HK	m_1	4 units
KA	m_2	4 units
HH	m_0	10 units
HHs	m_5	35 units

Table 5.1: Description of robot links

As discussed in the previous chapters, biped robot walk is considered as repetition of one step with two alternate phases: Single Support Phase (SSP) and Double Support Phase (DSP). In SSP, the stable foot holds the robot's weight while the swing foot is moving forward in the air. In DSP, both feet are on the ground.

5.2.2 Inverse and Forward Kinematics of Biped

In Figure 5.2, coordinate frames (x_i, y_i, z_i), $i = 0, 1, ..., 5$ are assigned at the joints starting from the stable leg's ankle to swing leg's ankle according to Denavit-Hartenberg (DH) procedure, and related joint angles θ_i; $i = 1, 2, ..., 5$ are demonstrated. The letter 'G' is assigned to indicate the universal coordinate frame. The numeric '0' is assigned to the base coordinate frame at the ankle joint of stable leg with axes x_0, y_0 and z_0 where x_0 is the direction of walking, z_0 is lying along the axis of rotation of joint 1 and y_0 is the axis according to the right hand thumb rule. The Denavit-Hartenberg (DH) parameters for the biped are given in Table 5.2.

Stable leg moves to follow a hip trajectory without knee bending with hip joint as end

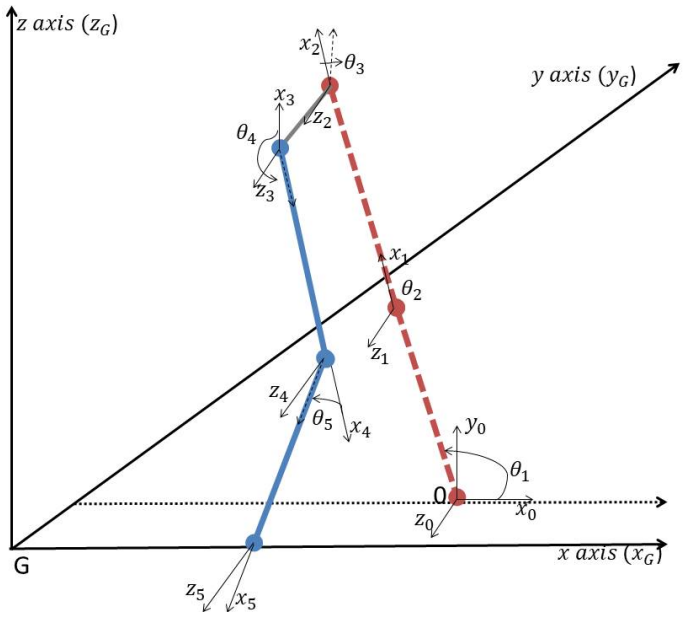


Figure 5.2: DH Model

Link	Joint angle(θ_i)	Twist angle(α_i)	Link length(a_i)	Joint length(d_i)
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	0	l_0
4	θ_4	0	l_2	0
5	θ_5	0	l_1	0

Table 5.2: DH parameters

effector and ankle joint as the base. Hence,

$$\theta_2 = 0 \tag{5.1}$$

${}^0Hs = ({}^0x_2, {}^0y_2, {}^0z_2)$ is the position coordinates of the stable leg's hip joint with respect to the base frame 0. Now

$${}^0Hs = {}^0T_G \ {}^GHs$$

where ${}^GHs = (x_H, l_0, z_H)$ is the position coordinates of hip joint Hs of the stable leg in frame G and the homogeneous transformation matrix representing the frame 0 with respect

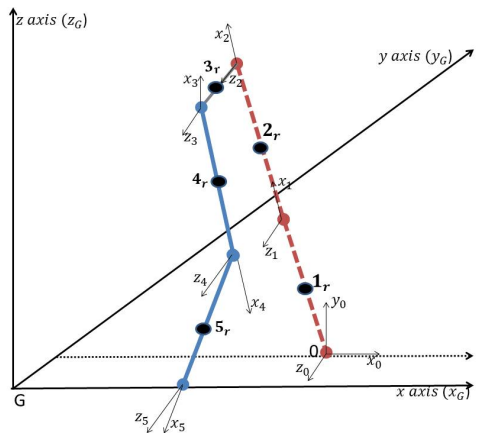


Figure 5.3: COM position in DH Structure

Link	${}^{i}r(r_{ix}, r_{iy}, r_{iz})$
1	${}^{1}r(-l_1/2, 0, 0)$
2	${}^{2}r(-l_2/2, 0, 0)$
3	${}^{3}r(0, 0, -l_0/2)$
4	${}^{4}r(-l_2/2, 0, 0)$
5	${}^{5}r(-l_1/2, 0, 0)$
joint angle	Joint angle limit
θ_1	$[-3\pi/4, -\pi/4]$
θ_2	[0,0]
θ_3	$[-5\pi/4, -\pi/2]$
θ_4	$[-\pi/4, \pi/2]$
θ_5	$[0, 3\pi/4]$

Table 5.3: COM of each link

to universal frame G is given by ${}^{G}T_0 = \begin{bmatrix} 1 & 0 & 0 & x_{As} \\ 0 & 0 & -1 & l_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

The joint angle θ_1 by treating stable leg as 1-link manipulator is given as:

$$\theta_1 = \tan^{-1}\left(\frac{{}^{0}y_2}{{}^{0}x_2}\right) \tag{5.2}$$

In Figure 5.2, axis z_3 and z_4 are collinear and according to DH algorithm x_4 axis can be taken in any perpendicular direction to this line. So, x_4 axis is taken in the direction of z_G for convenience. Now,

$$\theta_3 = \theta_1 - \pi/2; \tag{5.3}$$

By treating hip of swing leg as base and ankle joint as end effector, swing leg moves like 2 link manipulator. ${}^{3}A = ({}^{3}x_5, {}^{3}y_5, {}^{3}z_5)$ is the position of the swing leg's ankle joint which is expressed in terms of frame 3. Now

$${}^{3}A = {}^{3}T_G \ {}^{G}A$$

where ${}^{G}A = (x_A, 0, z_A)$ is the position of swing leg's ankle joint in frame G and the

transformation matrix from universal frame to frame 3 is ${}^{G}T_3 = \begin{bmatrix} 0 & 1 & 0 & x_H \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & z_H \\ 0 & 0 & 0 & 1 \end{bmatrix}$

The joint angles for this 2 link manipulator are

$$\theta_{1a} = \cos^{-1}\left(\frac{({}^{3}x_5)^2 + ({}^{3}y_5)^2 - {l_1}^2 - {l_2}^2}{2l_1 l_2}\right); \tag{5.4}$$

$$\theta_{2a} = \tan^{-1}\left(\frac{({}^{3}x_5)(l_1 + l_2\cos\theta_4) - ({}^{3}y_5)l_2\sin\theta_4}{({}^{3}y_5)(l_1 + l_2\cos\theta_4) + ({}^{3}x_5)l_2\sin\theta_4}\right) \tag{5.5}$$

As x_4 axis is in the direction of z_G and z_4 axis is in the direction of y_G, so it is easy to find the relation between θ_4 and θ_5 with θ_{1a} and θ_{2a}, which is given as

$$\theta_4 = \frac{\pi}{2} - \theta_{1a}; \tag{5.6}$$

$$\theta_5 = \theta_{2a}. \tag{5.7}$$

The coordinates ${}^{i}r = [r_{ix}, r_{iy}, r_{iz}]$, (in Figure 5.3) indicates the centers of mass with respect to the ith link frame $O_i(x_i, y_i, z_i); i = 1:5$ for the biped robot, are given in Table 5.3.

5.3 FNN and WNN Architecture for Trajectory Generation

In this section, FNN or WNN based approaches are proposed for trajectory generation. In this section, a general procedure for generating neural network based trajectory using given constraints is proposed. In next section, this procedure will be adapted for gait generation of the biped robot model.

5.3.1 Trajectory Generation for Given Conditions

For 4 Constraints

Let x(t) be an unknown trajectory which satisfies the following conditions:

$$x_A(t_0) = x_0, \;\; x_A(t_f) = x_p, \;\; \dot{x}_A(t_0) = x_{v_1}, \;\; \dot{x}_A(t_f) = x_{v_2}.$$

where x_0 is initial position, x_p is final position, x_{v_1} is initial velocity and x_{v_2} is final velocity. In order to generate this trajectory x(t) using FNN in the interval $t \in [t_0, t_f]$, the following

expression is considered

$$x(t) = \left(\frac{t-t_0}{t_f-t_0}\right)^2 N_1(t, W_1, V_1) + \left(\frac{t_f-t}{t_f-t_0}\right)^2 N_2(t, W_2, V_2), \tag{5.8}$$

where $N_i; i = 1, 2$ are neural networks with input t and adjustable weights W_i and V_i. W_i denotes the weight matrix from the input layer to hidden layer and V_i denotes the weight matrix from hidden layer to output layer. N_1 vanishes at initial time t_0 and N_2 vanishes at final time t_f. The architecture of NNs N_1 and N_2 to satisfy given constraints are given in Figures 5.4(a) and 5.4(b) respectively.

Further, in order to generate this trajectory x(t) using WNN in the interval $t \in [t_0, t_f]$, the following expression is considered:

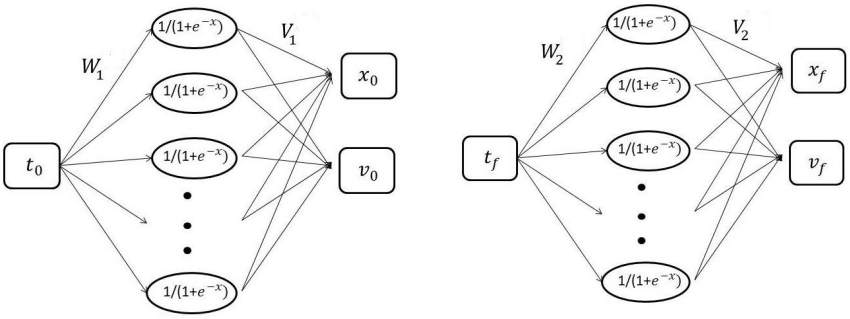
$$x(t) = \left(\frac{t-t_0}{t_f-t_0}\right)^2 N_{w1}(t, W_{w1}, V_{w1}, C_{w1}, D_{w1}) + \left(\frac{t_f-t}{t_f-t_0}\right)^2 N_{w2}(t, W_{w2}, V_{w2}, C_{w2}, D_{w2}) \tag{5.9}$$

where $N_{wi}; i = 1, 2$ are neural networks with input t. W_{wi} denotes the weight matrix from the input layer to hidden layer and V_{wi} denotes the weight matrix from hidden layer to output layer and C_{wi} and D_{wi} are translation and dilation vectors respectively in case of WNN. N_{w1} vanishes at initial time t_0 and N_{w2} vanishes at final time t_f.

Now, $x(t)$ is trained(in case of both the approaches) to satisfy all the constraints by minimizing the error E_1 (which is a function of the weights)

$$E_1 = (x(t_0) - x_0)^2 + (x(t_f) - x_p)^2 + \left(\frac{\partial x(t_0)}{\partial t} - x_{v_1}\right)^2 + \left(\frac{\partial x(t_f)}{\partial t} - x_{v_2}\right)^2 \tag{5.10}$$

The error E_1 is minimized by updating the weights using gradient decent approach as in equations 2.15 and 2.15 of Section 2.3 for FNN and in equations [2.18-2.20] of Section 2.4 for WNN in Chapter 2. Once the error reaches below a certain threshold value, the trained function acts as the required trajectory for the given boundary conditions. One network can satisfy two constraints easily but beyond that it is time consuming process. So extra neural networks are added and trained to satisfy other constraints.



(a) Structure of N_1 (b) Structure of N_2

Figure 5.4: FNN structure during weight training for x-t trajectory

For 6 Constraints:

Let x(t) be an unknown trajectory which satisfies the following conditions at the starting t_0, middle t_m and end points t_f respectively:

$$x(t_0) = x_0,\ \ x(t_f) = x_f,\ \ \dot{x}_A(t_0) = x_{v_1},\ \ \dot{x}_A(t_f) = x_{v_2}; x_A(t_m) = x_m,\ \ \dot{x}_A(t_m) = x_{v_3};$$

In order to generate this trajectory x(t) using NN in the interval $t \in [t_0, t_f]$, the following expression is considered:

$$x(t) = \left(\frac{t_m - t}{t_2}\right)^2 \left(\left(\frac{t - t_0}{t_f - t_0}\right)^2 N_1(t, W_1, V_1) + \left(\frac{t_f - t}{t_f - t_0}\right)^2 N_2(t, W_2, V_2)\right)$$

$$+ \left(\frac{(t_f - t)t}{t_m(t_f - t_m)}\right)^2 N_3(t, W_3, V_3) \qquad (Using\ \ FNN) \tag{5.11}$$

$$x(t) = \left(\left(\frac{t_m - t}{t_m}\right)^2 \left(\left(\frac{t - t_0}{t_f - t_0}\right)^2 N_{w1}(t, W_{w1}, V_{w1}, C_{w1}, D_{w1}) + \left(\frac{t_f - t}{t_f - t_0}\right)^2 N_{w2}(t, W_{w2},\right.\right.$$

$$\left.\left. V_{w2}, C_{w2}, D_{w2})\right) + \left(\frac{(t_f - t)t}{t_m(t_f - t_m)}\right)^2 N_{w1}(t, W_{w3}, V_{w3}, C_{w3}, D_{w3})\right) (Using\ \ WNN) \tag{5.12}$$

where, in FNN approach, N_1 vanishes at initial time t_0 and middle time t_m, N_2 vanishes at final time t_f and middle time t_m, and N_3 vanishes at initial time t_0 and final time t_f.

Similarly, in WNN approach, N_{w1} vanishes at initial time t_0 and middle time t_m, N_{w2} vanishes at final time t_f and middle time t_m, and N_{w3} vanishes at initial time t_0 and final time t_f.

Now, $x(t)$ is trained to satisfy all the constraints by minimizing the error E_2

$$E_2 = (x(t_0) - x_0)^2 + (x(t_f) - x_f)^2 + \left(\frac{\partial x(t_0)}{\partial t} - x_{v_1}\right)^2$$

$$+ \left(\frac{\partial x(t_f)}{\partial t} - x_{v_2}\right)^2 + (x(t_m) - x_m)^2 + \left(\frac{\partial x(t_m)}{\partial t} - x_{v_3}\right)^2$$

using gradient decent approach. Once the error reaches below a certain threshold value, the trained function acts as the required trajectory for the given boundary conditions.

5.3.2 Trajectory Modification for Changed Constraints using NN

Final Position or Velocity Changed

After training a trajectory for given constraints, if final position x_p and velocity x_{v_2} are changed to x_P and x_{V_2} during tracking at any intermediate time instant $t_s < t_f$ then the weights of N_2 are updated to satisfy x_F and x_{V_2} using the error function E_{1a}

$$E_{1a} = (x(t_f) - x_P)^2 + \left(\frac{\partial x(t_f)}{\partial t} - x_{V_2}\right)^2. \tag{5.13}$$

It takes very less time to train the weights for new constraints, as the initial weights used in the learning algorithm are the weights which were obtained for x_p and x_{v_2}.

Middle Position or Velocity Changed:

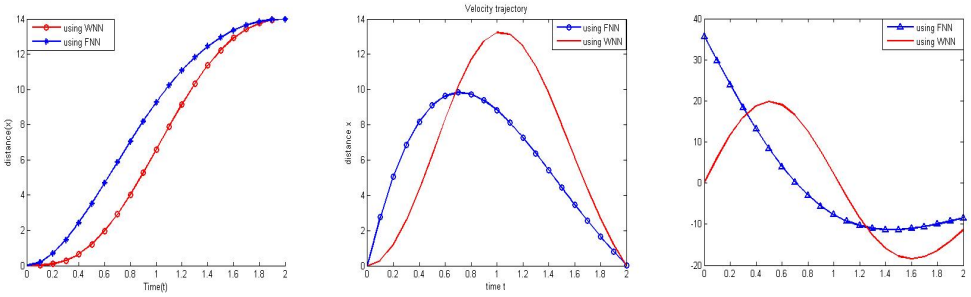
After training the weights for a trajectory, if middle position x_m and middle velocity x_{v_3} are changed to x_{mF} and x_{V_3} during trajectory tracking at any intermediate time instant $t_s < t_f/2$ then the weights of N_3 are updated to satisfy x_{mF} using the error function E_{2a}

$$E_{2a} = (x(t_m) - x_{mF})^2 + \left(\frac{\partial x(t_m)}{\partial t} - x_{V_3}\right)^2. \tag{5.14}$$

It takes very less time to train the weights for new constraints, as the initial weights used in the learning algorithm are the weights which are priorly trained for x_{mF} and x_{V_3}.

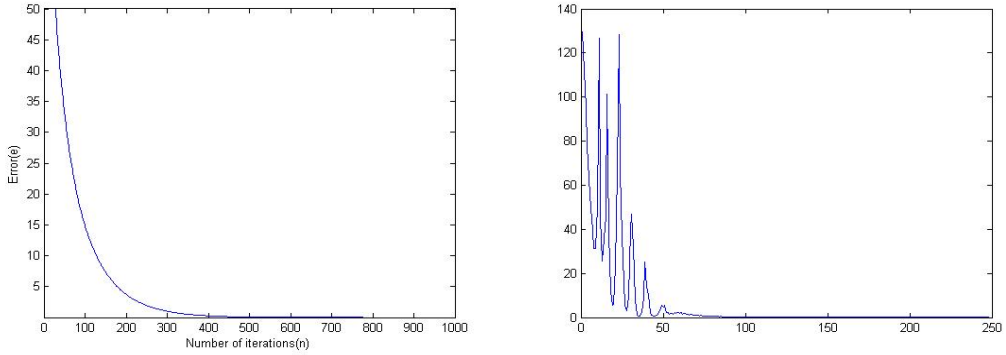
5.3.3 Results and Discussions of Proposed Approach

In this study, the learning rate (as mentioned in equations [2.15-2.20] for FNN are α_1=0.0001, α_2=0.0001 and the learning rate for WNN are α_1=0.0001, α_2=0.0001 α_3=0.01, α_4=0.01. The weights are initialized to random numbers between 0 and 1, translation vector C is initialized to zero vector and dilation vector D is initialized to unit vector. If the error reaches below a certain threshold (10^{-4}), then the FNN and WNN training stop.



(a) Position trajectories (b) Velocity trajectories (c) Acceleration trajectories

Figure 5.5: Proposed FNN verses WNN approach for (x_0=2, x_f=14, x_{v_1}=0, x_{v_2}=0)



(a) Error using FNN (b) Error using WNN

Figure 5.6: Errors for 4 constraints in trajectory generation

The results of proposed FNN and WNN approaches are compared for trajectory generation in Matlab2014a. In Figure 5.5(a), there are two trajectories red and black respectively, generated by using proposed FNN approach and WNN approach for 4 constraints. Figure 5.5 demonstrates that the trajectories and respective derivative trajectories are continuous and satisfy the given boundary conditions. The result of proposed approaches for trajectory

generation using 4, 6 and 8 constraints is given in Table 5.4. In this weight training algorithm, error converges smoothly to zero in less computation time which is shown in Table 5.4. After obtaining the FNN and WNN trajectory the robot can track the trajectory using the trained weights and these trajectory can adapt the change in constraints value quickly, also adaptation time for both are approximately same. Figures 5.6(a) and 5.6(b) shows the error graph for 4 constraints using FNN and WNN respectively. In Figure 5.6, WNN approach converges three times faster than FNN approach.

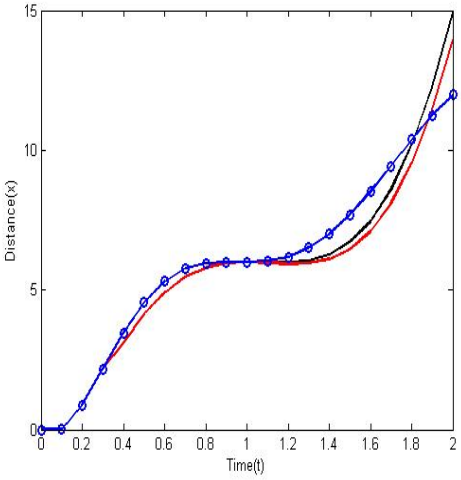
The blue line in Figures 5.7(a), 5.7(b) and 5.7(c) respectively, represents position, velocity

Boundary conditions	Values	Error	Convergence Time	
		E	For FNN	For WNN
$(x_0, x_f, x_{v_1}, x_{v_2})$	(2,14,0,0)	0.0001	15s	2.2s
$(x_0, x_f, x_{v_1}, x_{v_2}, x_m, x_{v_3})$	(2,14,4,-4,7,0)	0.0001	20s	5.13s
$(x_0, x_f, x_{v_1}, x_{v_2}, x_m, x_{v_3}, a_{c_1}, a_{c_2})$	(2,14,4,-4,7,0,0,0)	0.0001	22s	7s

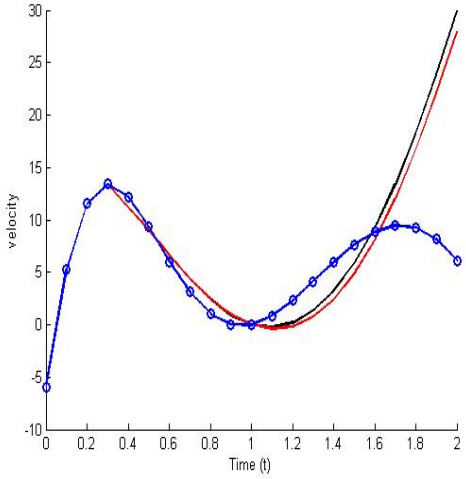
Table 5.4: Simulations based observations for Proposed approach

and acceleration trajectories based on FNN for 6 constraints (x_0=0, x_m=6, x_p=12, x_{v_1}=-6, x_{v_2}=6, x_{v_3}=0). Although, the blue line in Figures 5.8(a), 5.8(b) and 5.8(c) respectively, represents position, velocity and acceleration trajectories based on WNN for same constraints (as for FNN). If for the given trajectories using FNN and WNN, final constraints values are changed (during tracking) to x_P=14, x_{V_2}=0 at time instant t=0.4 sec and to x_P=15 , x_{V_2}=0 at time instant t=0.9 sec then the respective changes in trajectory and derivative trajectory are shown by red line and black line in the same Figures ((5.7),(5.8)). Also for the same constraints, if middle position x_m is changed (during tracking) from 6 to 7 units at time instant t=0.4 sec then red line in Figures 5.9 and 5.10 represents the modified trajectory using FNN and WNN respectively. Similarly, blue line in Figures 5.11 and 5.12 respectively, represents trajectories based on FNN and WNN for 6 constraints (x_0=3,x_m=6, x_p=3, x_{v_1}=-4, x_{v_2}=4,x_{v_3}=0) and in same figures, red line shows modified trajectories for a change in final constraints x_p=3, x_{v_1}=-4, to x_P=12.5, x_{V_2}=0 at time instant t=0.4 sec.

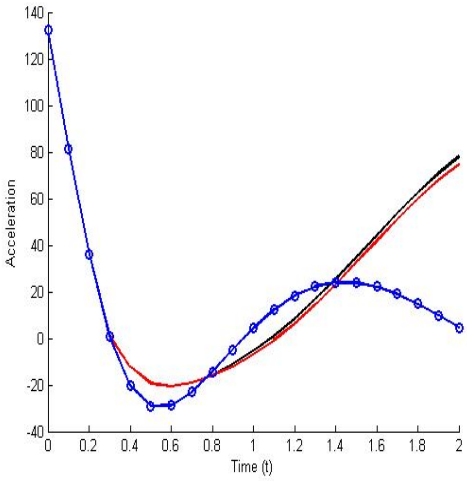
So it can be concluded that FNN and WNN approaches can adapt the changes in constraint values during tracking and modify the trajectory accordingly. Although WNN convergence is 3 time faster than FNN, FNN trajectory is more smooth in comparison with WNN trajectory which is an important aspect for stable walk.



(a) Position trajectories

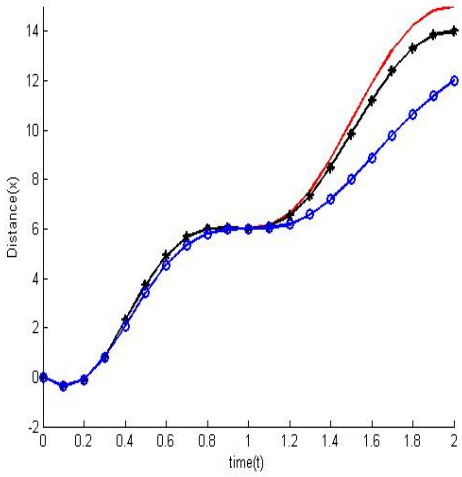


(b) Velocity trajectories

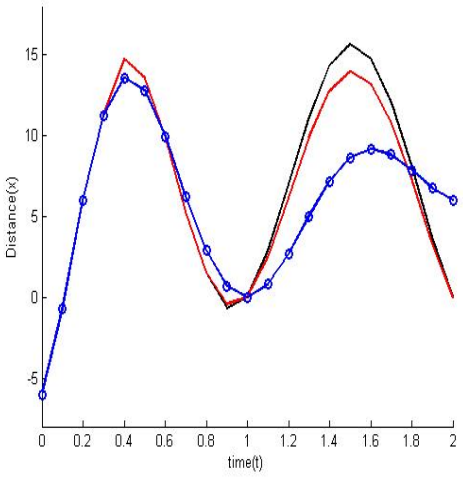


(c) Acceleration trajectories

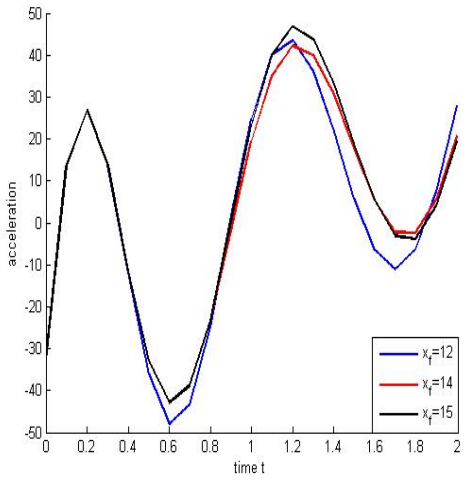
Figure 5.7: Trajectory using proposed FNN approach which is modified for final conditions at time t_s=0.4 to x_P=14, x_{V_2}=0 and at t=0.9 to x_P=15 , x_{V_2}=0



(a) Position trajectories

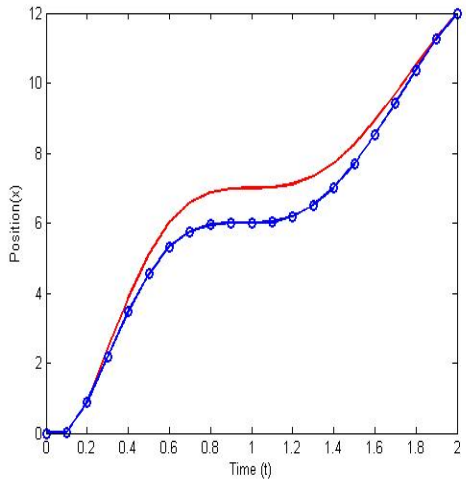


(b) Velocity trajectories

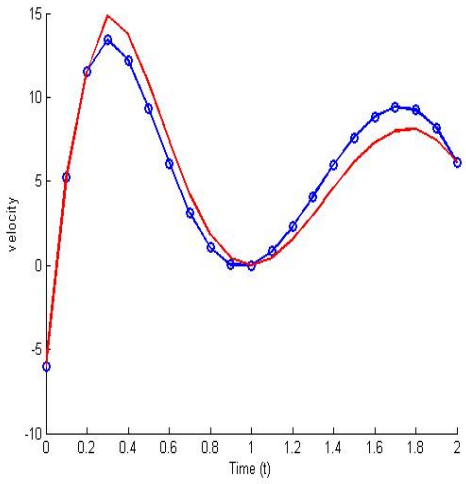


(c) Acceleration trajectories

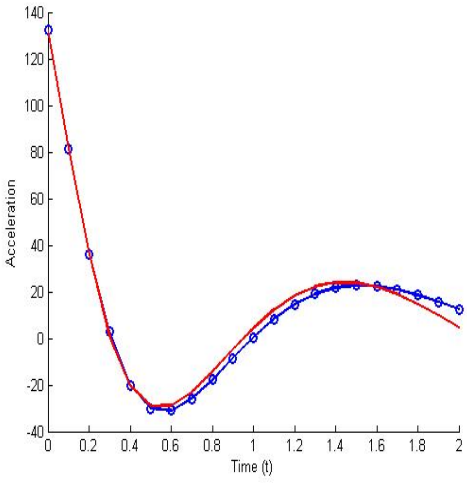
Figure 5.8: Trajectory using proposed WNN approach which is modified for final conditions at time t_s=0.4 to x_p=14, x_{v_2}=0 and at t=0.9 to x_P=15 , x_{V_2}=0



(a) Position trajectories

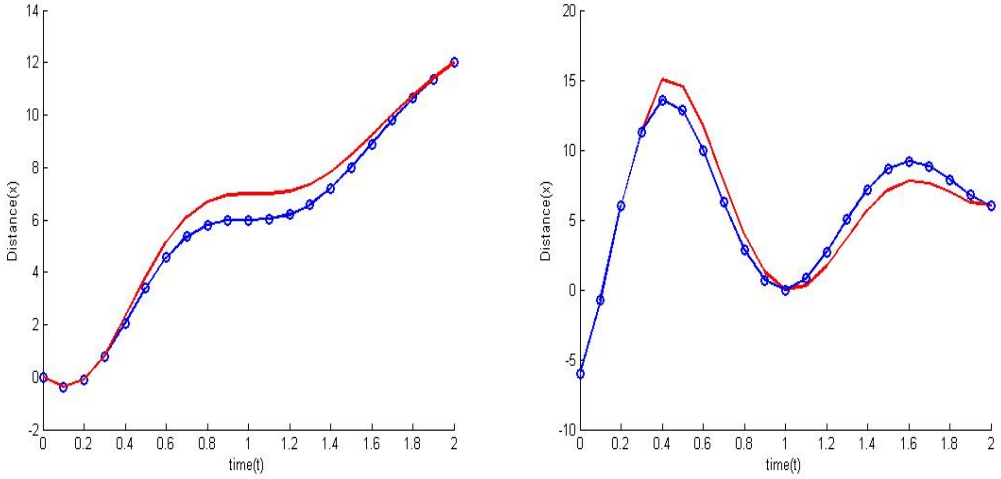


(b) Velocity trajectories



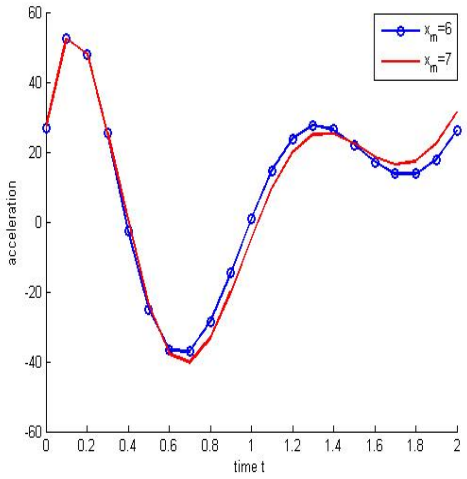
(c) Acceleration trajectories

Figure 5.9: FNN trajectory for (x_0=0,x_m=6, x_p=12, x_{v_1}=-6, x_{v_2}=0,x_{v_3}=6) which modified at $t_s = 0.4$ for changed middle position x_m=7



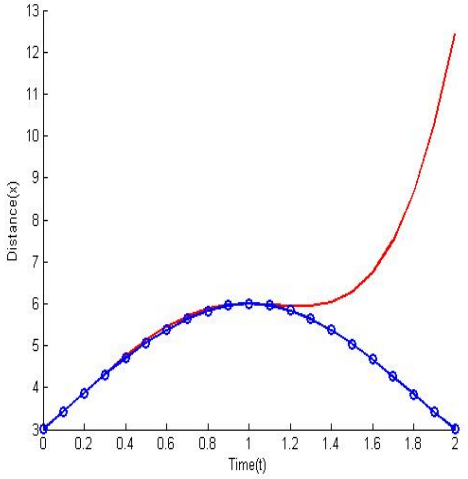
(a) Position trajectories

(b) Velocity trajectories

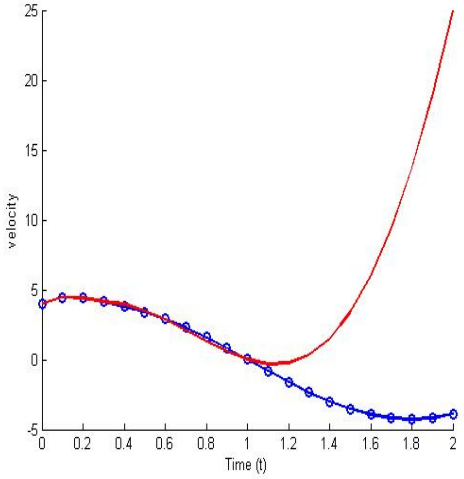


(c) Acceleration trajectories

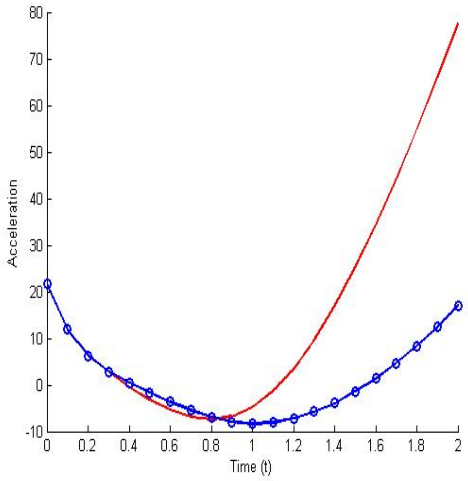
Figure 5.10: WNN trajectory for (x_0=0,x_m=6, x_p=12, x_{v_1}=-6, x_{v_2}=0,x_{v_3}=6) which modified at $t_s = 0.4$ for changed middle position x_m=7, x_{V_3}=0



(a) Position trajectories

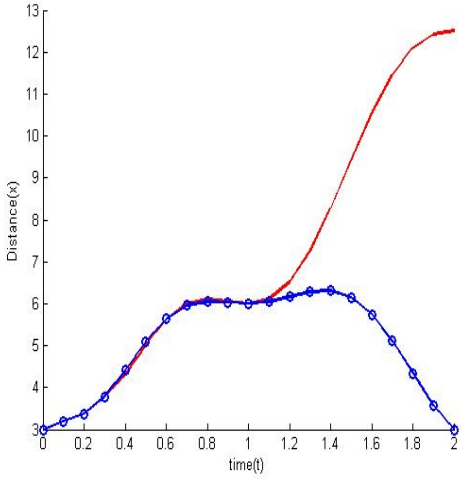


(b) Velocity trajectories

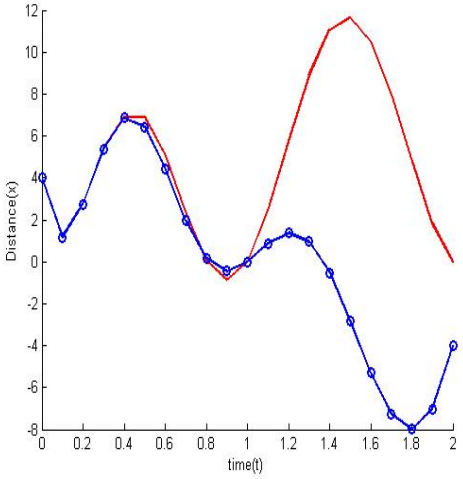


(c) Acceleration trajectories

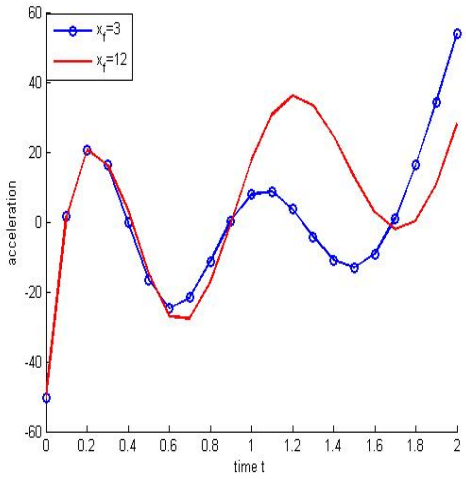
Figure 5.11: FNN trajectory for(x_0=3,x_m=6, x_p=3, x_{v_1}=4, x_{v_3}=0,x_{v_2}=-4)which modified for changed final position x_P=12.5,x_{V_2}=0



(a) Position trajectories



(b) Velocity trajectories



(c) Acceleration trajectories

Figure 5.12: WNN trajectory for(x_0=3,x_m=6, x_p=3, x_{v_1}=4, x_{v_2}=0,x_{v_3}=-4)which modified for changed final position x_P=12.5,x_{V_2}=0

5.4 Gait Generation

The hip, ankle and upper body trajectories are the parts of biped's Gait. The swing leg

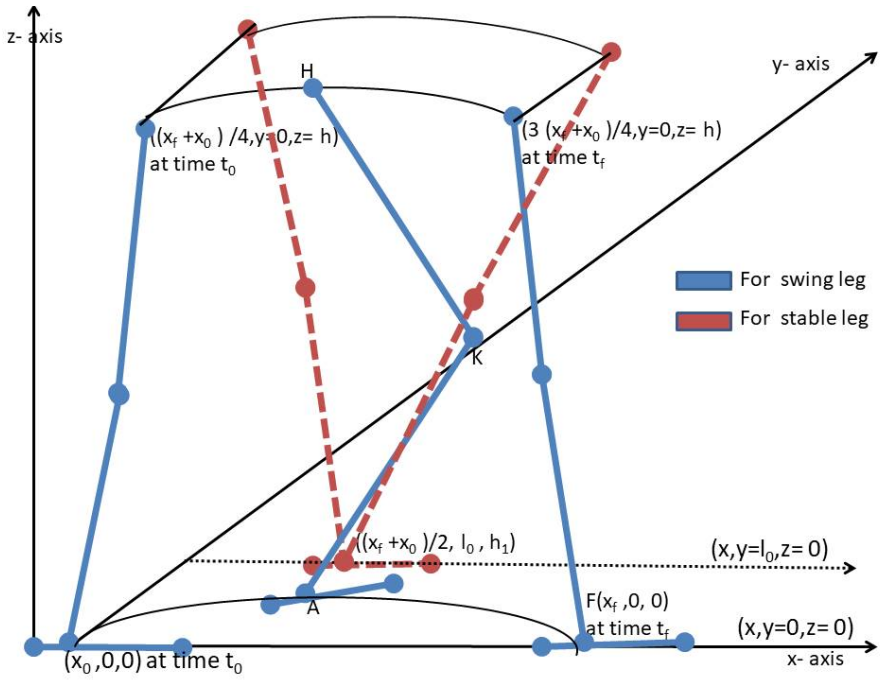


Figure 5.13: Biped robot's walk

moves forward during SSP and becomes the stable leg in upcoming step. Trajectories have been generated, for the swing leg's ankle and hip joints (both hips follow same trajectory) in the sagittal plane $(X_G - Z_G)$ and for upper body in frontal plane $(Y_G - Z_G)$. For hip and ankle trajectories planning, x-trajectory is generated in terms of time t and z-trajectory is generated in terms of distance x.

5.4.1 Ankle Trajectory

Let $(x_A(t), 0, z_A(t)) : t \in [t_0, t_f]$ be the coordinate of the swing leg's ankle joint. The x-trajectory for ankle joint A must satisfy the following position and velocity constraints:

$$x_A(t_0) = x_0, \;\; x_A(t_f) = x_p, \;\; \dot{x}_A(t_0) = 0, \;\; \dot{x}_A(t_f) = 0;$$

The z-trajectory for ankle joint A as a function of x must satisfy the following constraints:

$$z_A(x_0) = 0, \;\; z_A(x_p) = 0, \;\; z_A(x_m) = h_1, \;\; \dot{z}_A(x_m) = 0;$$

where $x_p = x_0 + x_f$ with step length x_f, h_1 is step height, $x_m = x_A(t_f/2)$ is the middle position of x_0 and x_f as given in Figure 5.13.

To satisfy these constraints, x_A can be generated by using equation (5.8) in Section 5.3 (with error function E_1) and similarly we can define the following z-trajectory:

$$\begin{aligned} z_A(x_A) \;\; = \;\; & \left(\frac{x_m - x_A(t)}{x_m}\right)^2 \left(\left(\frac{x_p - x_A(t)}{x_p - x_0}\right) N_5(x_A(t), W_5, V_5) \right. \\ & \left. + \left(\frac{x_A(t) - x_0}{x_p - x_0}\right) N_6(x_A(t), W_6, V_6)\right) \\ + \;\; & \left(\frac{x_A(t)(x_p - x_A(t)}{x_m(x_p - x_m)}\right) N_7(x_A(t), W_7, V_7); \end{aligned} \tag{5.15}$$

The error function which is to be minimized by updating weights in networks N_5, N_6 and N_7 for z-trajectory is given by

$$E_3 = (z_A(x_0))^2 + (z_A(x_p))^2 + (z_A(x_m) - h_1)^2 + \left(\frac{\partial z_A(x_m)}{\partial t}\right)^2$$

The weights are updated by minimizing the error using gradient descent approach.

5.4.2 Hip Trajectory

Both the hips follows the same trajectory in xz plane.

For stable leg, the hip follows a circular trajectory with radius $(l_1 + l_2)$ and center at its ankle joint. Boundary conditions related to hip coordinate $(x_H(t), y_H(t), z_H(t))$ are given by the following:

For motion in x-direction:

$$x_H(t_0) = x_0 + x_f/4, \;\; x_H(t_f) = x_0 + 3x_f/4, \;\; \dot{x}_H(t_0) = v_3, \;\; \dot{x}_H(t_f) = v_4;$$

For motion in z-direction:

$$z_H(x_0) = h, \;\; z_H(x_p) = h, \;\; \dot{z}_H(x_0) = 0, \;\; \dot{z}_H(x_p) = 0;$$

where h is maximum hip height (for both initial and final position), v_3 and v_4 are the initial and final velocities of hip in x-direction respectively.

The x-trajectory can be calculated by the Equation (5.8) and weights are trained using error function E_4 given by

$$E_4 = (x_H(t_0) - (x_0 + x_f/4)^2 + (x_H(t_f) - x_0 + 3x_f/4)^2 + \left(\frac{dx_H(t_0)}{dt} - v_3\right)^2 + \left(\frac{dx_H(t_f)}{dt} - v_4\right)^2$$

and the z-trajectory is defined as

$$z_H(x_H) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_0 + x_f/2))^2}; \tag{5.16}$$

5.4.3 Upper Body Trajectory

The upper body mass has its effect on biped's stability. For this, suitable choice of the mass and position, velocity and acceleration conditions of the upper body will provide better walking efficiency.

The stable leg balanced the whole body weight during SSP. U starts to move in y-direction from middle point of hips to the stable leg side in half step time($t_2 = t_f/2$) and moves back to middle point of hips in rest of the time (as in Section 3.6 of chapter 3). So the boundary conditions for the upper body are

$$y_U(t_0) = y_l; y_U(t_m) = y_a; y_U(t_f) = y_l;$$
$$\frac{\partial y_U(t_0)}{\partial t} = v_5; \frac{\partial y_U(t_m)}{\partial t} = 0; \frac{\partial y_U(t_f)}{\partial t} = v_6; \frac{\partial^2 y_U(t_0)}{\partial t^2} = 0; \frac{\partial^2 y_U(t_f)}{\partial t^2} = 0;$$

where $y_l = l_0/2$ is the middle point of hips, v_7 and v_8 are the initial and final upper body velocity, y_a is the upper body position at initial and final time. Then, the upper body (U) trajectory in y-direction is

$$y_U(t) = x(t) + \left(\frac{(t - t_m)(t_m - t)}{(t_f - t_0)}\right)^2 \left(\frac{(t - t_0)}{t_f - t_0}\right)^2 N_8(t, W_8, V_8) + \left(\frac{(t_f - t)}{t_f - t_0}\right)^2 N_9(t, W_9, V_9) \tag{5.17}$$

The FNN networks N_8 and N_9 are added to $x(t)$ (as in equation (5.8)) to satisfy the acceleration constraints at initial time t_0 and at final time t_f. The weights are updated by minimizing the error

$$E_5 = \left(\frac{d^2 y_U(t_0)}{dt^2}\right)^2 + \left(\frac{d^2 y_U(t_f)}{dt^2}\right)^2 \tag{5.18}$$

Similarly, all joint trajectories can be generated by using WNN.

5.4.4 Simulated Results

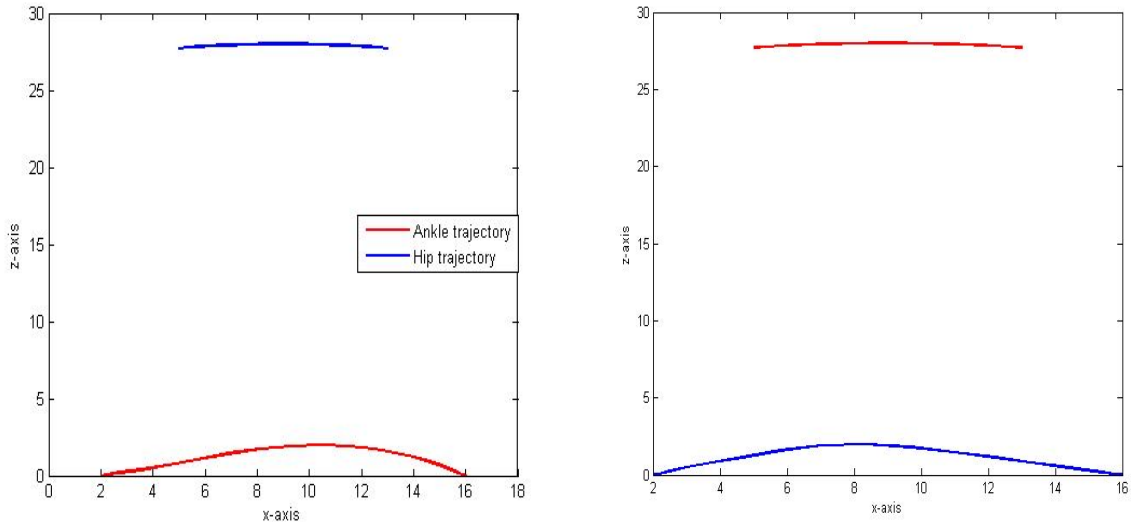
Parameters used for simulation of the trajectories are given in Table-5.5.
Initially, swing foot and stable foot are on the ground and they are positioned at $0 < x < 5, -2 < y < 2$ and $7 < x < 12, 6 < y < 10$ respectively.

	Parameter	Value
For step	x_f	16 units
	h_1	2 units
	t_0	0 sec
	x_0	2 units
	t_f	2 sec
	l_0	8 units
For hip	v_3	4.2 units
	v_4	-4.2 units
For upper body	y_a	7 units
	v_5	5.4 units
	v_6	-5.4 units
Learning rate	α	0.0001
Hidden layer neurons in NNs	For 4 constraints	30
	For 6 constraints	45
	For 8 constraints	75

Table 5.5: Parameters

Figures 5.14(a) and 5.14(b) respectively, represents the desired ankle and hip trajectories in x-z plane using proposed FNN and WNN approaches. Whole body's COM trajectory is given in Figure 5.15(a). The inverse kinematic solutions(with respect to time) for hip and ankle trajectories are shown in Figure 5.15(b).

The upper body has its effect on ZMP stability. FNN and WNN trajectories are given in Figure 5.16(a) and respective acceleration trajectories are given in Figure 5.16(b). In Figure 5.16(b), upper body acceleration is less smoother in comparison of WNN. So, proposed WNN approach is not suitable for upper body motion. The ZMP graphs generated by using FNN(blue) and WNN(red) approaches for ankle trajectory are given in Figures 5.17. From these figures, it can be concluded that the stability margin is high for FNN trajectory. Hence, FNN approach is preferable over WNN.



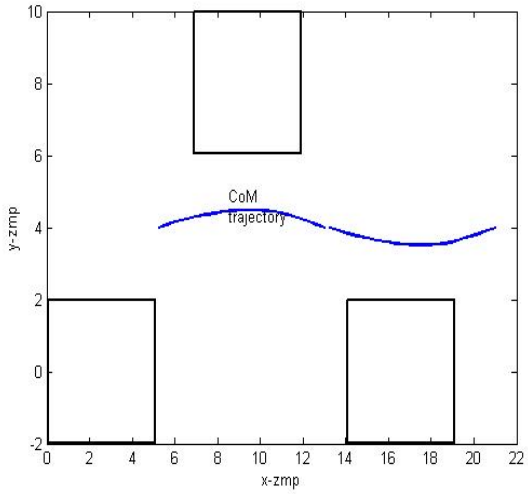
(a) Trajectory generation using FNN

(b) Trajectory generation using WNN

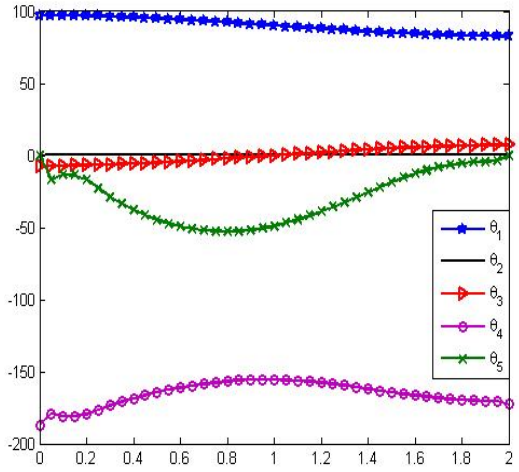
Figure 5.14: Ankle and Hip trajectories in xz-plane

5.5 Conclusion

To generate a trajectory, the conventional approach is to fit a polynomial by satisfying the given conditions. However, NN provides a better alternative for trajectory generation. In this chapter, novel FNN and WNN approaches are proposed for trajectory planning to produce smooth motion and the performance of WNN and FNN are compared. The approaches have been tested for 4 and 6 constraints yielding good results in simulation and can be modified at any instant of time during tracking. Although WNN has fast convergence to trained the modeled trajectory as compared to FNN, it is not found to be suitable for biped walking trajectory generation. The reason is the acceleration trajectory by using WNN approach is not smooth as compare to FNN (see Figure 5.17). But if the foot size is large enough then WNN is preferable to generate the ankle trajectory faster. Both the approaches can adapt the changes in constraints due to environmental disturbances, obstacles and sensor noise etc during tracking which is discussed in next chapter and adaptation time is same for both the approaches.

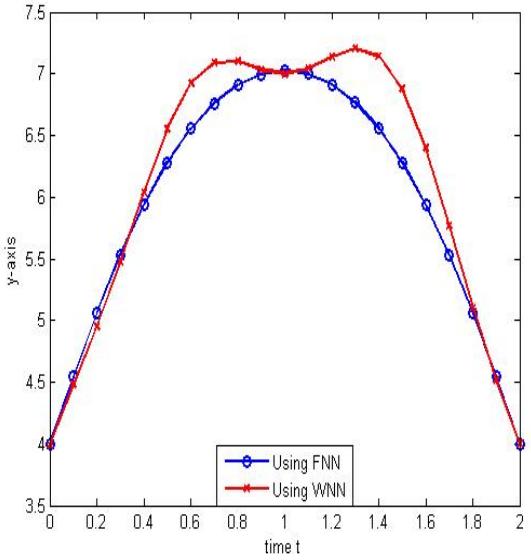


(a) COM trajectory

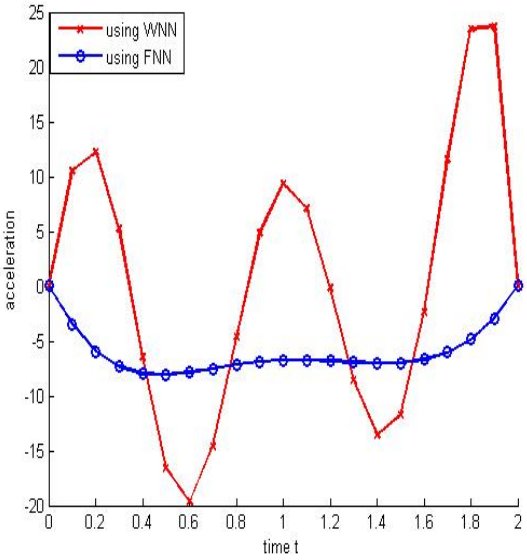


(b) Joint angles of biped robot

Figure 5.15:



(a) FNN vs WNN position trajectories



(b) FNN vs WNN acceleration trajectories

Figure 5.16: Upper body

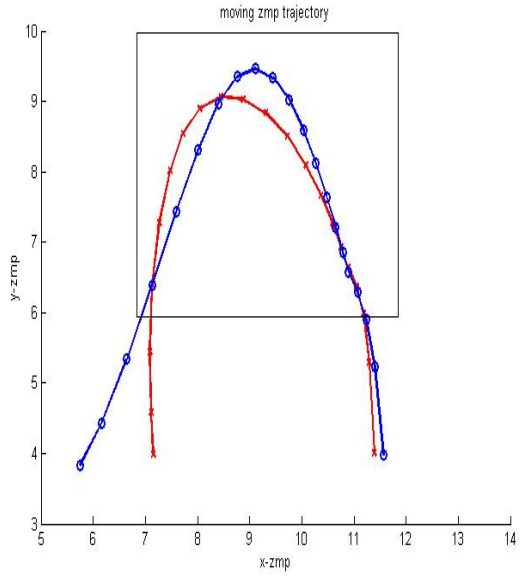


Figure 5.17: ZMP for FNN vs WNN ankle trajectory

Chapter 6: Walking Control of Biped Robot using FNN

In this chapter, the dynamic equation of a 5 DOF Biped Robot model (given in previous chapter) is derived with the assumption that a mass is moving along the hip during the walk to maintain the stability. Trajectory planning is the main concern of biped walking to avoid obstacles and ditch in an uncertain environment whose position and the shapes are not known a priori. A FNN based trajectory generation approach is proposed for the given boundary conditions in the previous chapter. In this chapter, we verified the ability of biped using FNN approach to adjust its gait/step according to uncertain environment during tracking. The suitability of the proposed approach for biped robot is studied using Zero Moment Point (ZMP) stability criteria and PD controller. The results are compared and simulated using Matlab2014a which show the effectiveness of the proposed approaches.

6.1 Introduction

Biped dynamics with controller plays an important role for the stable locomotion [7, 11, 41, 45, 81, 98]. One of the great challenges for a biped is to walk in an unknown environment/terrain. Recently, various techniques have been extensively studied in the literature [15, 51, 72, 90, 104] for this purpose. Some robots are also able to cross the obstacles of different heights [15, 51, 72, 90, 104]. Stasse et al. [90] presented a HRP-2 humanoid robot that can dynamically step over a 15 cm obstacle within 4 secs using quasi static stability with 3 cm stability margin. On the other hand, Li et al. [51] planned a gait by motion capture system towards overcoming the obstacle during walk of the humanoid robot. Force sensors are used to calculate the ground reaction forces. FNN based optimization procedure is used to calculate the optimum coefficient of polynomial for given constraints on

uneven surface and to cross a ditch [26, 97]. However, these methods don't work well in more complex domain as the trajectories can not modified during walking according to terrain conditions. Humans do not walk based on a predefined trajectory rather they modify the trajectory during tracking when the environment/terrain conditions change possibly to avoid obstacles, to cross a ditch and to adapt disturbance in parameters due to sensor noise etc. So the biped should walk in a similar manner. Biped should have a better ability to move on an uneven terrain where a small unexpected disturbance can cause robot to fall [34, 42]. Hence, the development of bipedal robot which can adjust its gait/step according to terrain during tracking quickly is current field of research. NN can be effective in generating a trajectory for this purpose.

In this chapter, semi-supervised FNN based approach as proposed in previous chapter is used to generate walk pattern for 5 DOF biped robot and results are compared with basic polynomial approach. The trained trajectory can be modified for updated value of constraints during tracking quickly as it takes very less time to train the weights because the initial weights used in the learning algorithm (for the new trajectory) are the weights which were priorly trained(for the old trajectory). ZMP stability criteria is used to ensure the dynamic balance for these NN trajectories. PD controller is used to control the motion.

The rest of paper is organized as follows. Dynamic equation of this robot model is given in Section 6.2. Gait planning procedure for feet, hip and upper body is described in Section 6.3. Section 6.4 is dedicated to results and discussions. Section 6.5 presents the conclusion.

6.2 Dynamic Equations of Proposed Biped Model

Dynamic equation of motion describes the relationship of the actuation and contact forces with the acceleration and motion trajectories which are due to the reacting forces. For the motion of each joint, torque/force is required to follow a desired trajectory. There are two main method to derive the dynamic equation of a robot, namely, Newton-Euler and Euler-Lagrange methods. Here, we follow the Lagrange-Euler method to derive the equation.

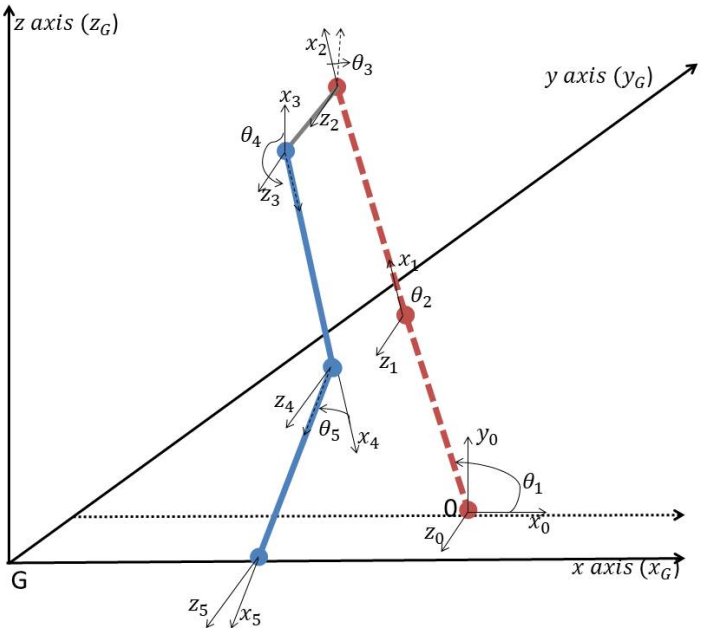


Figure 6.1: DH Structure

6.2.1 Forward Kinematics

In Figure 6.1, coordinate frames (x_i, y_i, z_i), $i = 0, 1, 2....5$ are assigned at the joints starting from the stable leg's ankle to swing leg's ankle according to Denavit-Hartenberg (DH) procedure(as in Section 2.2.2 in Chapter 2), and related joint angles θ_i; $i = 1, 2, ...5$ are demonstrated. The letter 'G' is assigned to indicate the universal coordinate frame. The numeric '0' is assigned to the base coordinate frame at the ankle joint of stable leg with axes x_0, y_0 and z_0 where x_0 is the direction of walking, z_0 is lying along the axis of rotation of joint 1 and y_0 is the axis according to the right hand thumb rule. The DH parameters for the biped are given in Table 6.1.

Link	Joint angle(θ_i)	Twist angle(α_i)	Link length(a_i)	Joint length(d_i)
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	0	l_0
4	θ_4	0	l_2	0
5	θ_5	0	l_1	0

Table 6.1: DH parameters

By using these DH parameters, the homogeneous transformation matrix ${}^{i-1}T_i$ (as in Section 2.1 of Chapter 2) are expressed as:

$${}^{0}T_1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & l_1C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & l_1S\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{1}T_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & l_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & l_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2}T_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & 0 \\ S\theta_3 & C\theta_3 & 0 & 0 \\ 0 & 0 & 1 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{3}T_4 = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & l_2C\theta_4 \\ S\theta_4 & C\theta_4 & 0 & l_2S\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{4}T_5 = \begin{bmatrix} C\theta_5 & -S\theta_5 & 0 & l_1C\theta_5 \\ S\theta_5 & C\theta_5 & 0 & l_1S\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $S\theta_i = sin\theta_i$, $C\theta_i = cos\theta_i$.
The forward kinematics of this biped robot can be derived by using the frame transformations from the stable leg's ankle joint to swing leg's ankle joint, which is given by:

$${}^{0}T_5 = \prod_{i=1}^{5} {}^{i-1}T_i$$

6.2.2 Joint Velocities of a Robot

Let ${}^{i}r$ be a point on the ith link and expressed in homogeneous coordinates with respect to the ith joint coordinate frame as

$${}^{i}r = (r_{ix}, r_{iy}, r_{iz}, 1)^T$$

The coordinate of the centers of masses(COM) for all the 5 link are given in the Table 6.2.

Link	coordinate of COM ${}^{i}r(r_{ix}, r_{iy}, r_{iz})$
1	${}^{1}r(-l_1/2, 0, 0)$
2	${}^{2}r(-l_2/2, 0, 0)$
3	${}^{3}r(0, 0, -l_0/2)$
4	${}^{4}r(-l_2/2, 0, 0)$
5	${}^{5}r(-l_1/2, 0, 0)$

Table 6.2: COM of each link

Let $Q_i; i = 1, 2, ..., 5$ be a matrix defined as

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

As all the joints are revolute, the partial derivative of ${}^{i-1}T_i$ with respect to θ_i can be easily calculated with the help of Q_i as

$$\frac{\partial^{i-1}T_i}{\partial\theta_i} = Q_i \ {}^{i-1}T_i$$

Let U_{ij} andU_{ijk} be defined as:

$$U_{ij} \cong \frac{\partial^0 T_i}{\partial\theta_j} = \begin{cases} {}^0T_{j-1} \ Q_j \ {}^iT_{j-1} & j \leq i \\ 0 & j > i \end{cases}$$

$$U_{ijk} \cong \frac{\partial U_{ij}}{\partial\theta_k} = \begin{cases} {}^0T_{j-1} \ Q_j \ {}^{j-1}T_{k-1} \ Q_k \ {}^{k-1}T_i & i \geq k \geq j \\ {}^0T_{k-1} \ Q_k \ {}^{k-1}T_{j-1} \ Q_j \ {}^{j-1}T_i & i \geq j \geq k \\ 0 & i < j \ \ or \ \ i < k \end{cases}$$

For example $U_{11} = \frac{\partial^0 T_1}{\partial\theta_1} = Q_1 {}^0T_1, \ \ U_{12} = \frac{\partial^1 T_2}{\partial\theta_2} = Q_2 {}^1T_2,$

$\frac{\partial U_{12}}{\partial\theta_1} = Q_1 Q_2 {}^1T_2,$

Using these notations velocity v_i of the point ${}^i r$, expressed in the base coordinate frame is

$$\begin{aligned} v_i &= \frac{d}{dt}({}^0T_i\, {}^i r) \\ &= \left[\sum_{p=1}^{i} \frac{\partial {}^0T_i}{\partial \theta_p}\dot{\theta}_p\right] {}^i r = \sum_{p=1}^{n} U_{ip}\dot{\theta}_p\, {}^i r. \end{aligned}$$

6.2.3 Euler-Lagrange Equation

Let K and P be the total kinetic and potential energy respectively for the biped. The Lagrangian is defined as

$$L = K - P$$

The total kinetic energy K is

$$K = 1/2\sum_{i=1}^{5}\sum_{p=1}^{i}\sum_{q=1}^{i}\left[Tr(U_{1p}J_iU_{1q}^T)\dot{\theta}_p\dot{\theta}_q\right];$$

where the pseudo inertia matrices $J_i; i = 1, 2..., 5$ (assuming that all the products of inertia are zero) are

$$J_1 = \begin{bmatrix} 1/3m_1l_1^2 & 0 & 0 & -1/2m_1l_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2m_1l_1 & 0 & 0 & m_1 \end{bmatrix} J_2 = \begin{bmatrix} 1/3m_2l_2^2 & 0 & 0 & -1/2m_2l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2m_2l_2 & 0 & 0 & m_2 \end{bmatrix}$$

$$J_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & m_0l_0^2/3 + m_5l(t)^2 & m_0l_0/2 + m_5l(t) \\ 0 & 0 & m_0l_0/2 + m_5l(t) & m_0 + m_5 \end{bmatrix}$$

$$J_4 = \begin{bmatrix} 1/3m_3l_2^2 & 0 & 0 & -1/2m_3l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2m_3l_2 & 0 & 0 & m_3 \end{bmatrix} J_5 = \begin{bmatrix} 1/3m_4l_1^2 & 0 & 0 & -1/2m_4l_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2m_4l_1 & 0 & 0 & m_4 \end{bmatrix}$$

where $m_i; i = 1, 2, ..., 5$ are the masses of links, M is moving mass on the hip and l(t) is distance between the moving mass and the COM of hip. The total potential energy P of the biped robot is

$$P = \sum_{i=1}^{n}(-m_i g^0 T_i^i \bar{r}_i) \;\; i = 1, 2, ..., 5 \tag{6.1}$$

where gravity row vector $\mathbf{g} = (0, -|g|, 0, 0)$ is expressed in the base coordinate system with the gravitational constant g.

Then, the lagrangian function L is given by

$$L = 1/2 \sum_{i=1}^{n}\sum_{p=1}^{i}\sum_{q=1}^{i}\left[Tr(U_{ip} J_i U_{iq}^T)\dot{\theta}_p \dot{\theta}_q\right] + \sum_{i=1}^{n}(m_i g^0 T_i^i \bar{r}_i) \tag{6.2}$$

Applying the Lagrange-Euler formulation to the lagrangian function of the biped robot, the necessary generalized torque τ_i for joint i [20]is

$$\tau_i = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_i}\right] - \frac{\partial L}{\partial \theta_i}; \;\; i = 1, 2..., n; \;\; n = 5.$$

$$\tau_i = \sum_{p=1}^{n}\sum_{q=1}^{p} Tr(U_{pq} J_j U_{pi}^T)\ddot{\theta}_q + \sum_{p=1}^{n}\sum_{q=1}^{p}\sum_{m=1}^{p} Tr(U_{pqm} J_p U_{pi}^T)\dot{\theta}_q \dot{\theta_m} - \sum_{p=1}^{n} m_p g(U_{pi} \;\; {}^p\bar{r}_p)$$

The above equations can be expressed in matrix form as

$$\tau(t) = D(\theta(t))\ddot{\theta}(t) + C(\theta(t), \dot{\theta}(t)) + G(\theta(t)) \tag{6.3}$$

where

$\tau(t) = n \times 1$ generalized torque vector applied at joints.

$\theta(t) = n \times 1$ joint variables vector.

$\dot{\theta}(t) = n \times 1$ joint velocity vector.

$\ddot{\theta}(t) = n \times 1$ joint acceleration vector.

$D(\theta) = n \times n$ inertial acceleration related symmetric matrix.

$C(\theta, \dot{\theta}) = n \times 1$ nonlinear Coriolis and centrifugal force vector.

$G(\theta) = n \times 1$ gravity force vector.

Hence, the joint torques (τ_i; i=1,2,...,5) can be determined as follows:

$$\tau_i = \sum_{q=1}^{n} D_{iq}\ddot{\theta}_q + C_i(\theta_i, \dot{\theta}_i) + G_i(\theta(t)) \tag{6.4}$$

where inertia, gravity and Coriolis/centrifugal terms respectively are given by

$$D_{iq} = \sum_{p=max(i,q)}^{5} Tr(U_{pq} J_p U_{pq}^T) \;\; i,q = 1,2,...,5 \tag{6.5}$$

$$G_i = \sum_{p=1}^{5} (-m_p g U_{pi} \;\; {}^p\bar{r}_p) \;\; i = 1,2,...,5. \tag{6.6}$$

$$C_i = \sum_{q=1}^{5}\sum_{m=1}^{5} h_{iqm}\dot{\theta}_q\dot{\theta}_m \;\; i = 1,2,...,5 \tag{6.7}$$

with $c_{iqm} = \sum_{p=max(i,q,m)}^{n} Tr(U_{pqm} J_p U_{pi}^T) \;\; i,q,m = 1,2,...,5.$
The expressions for D_{iq} , H_i and C_i $i = 1,2,...,5, q = 1,2,...,5$ are given at the end of this chapter.

PD Control

The input torque for each joint is

$$\tau_i = K_{pi}(\theta_{di} - \theta_i) + K_{di}(\dot{\theta}_{di} - \dot{\theta}_i) \tag{6.8}$$

where, θ_{di} is the desired joint angle, $\dot{\theta}_{di}$ is the desired angular velocity for each joint $i = 1,2,...5$. K_{pi} and K_{di} are the coefficients of the proportional and differential terms respectively.

As the double support phase in case of flat footed robot is instantaneous, walk is only considered for single support phase in this chapter.

6.3 Gait Generation

The hip, ankle and upper body trajectories are the parts of biped's Gait. These trajectories are generated using two approaches namely, FNN approach and polynomial approach.

6.3.1 Using Proposed FNN Approach

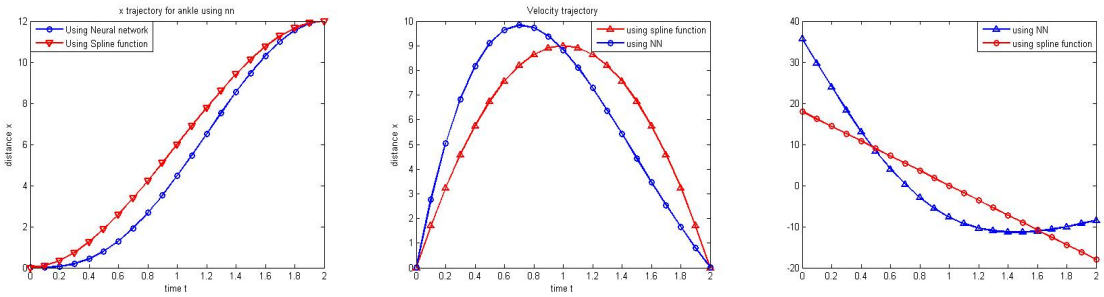
In the FNN approach the trail function is defined using FNN where one network N_i is defined for each ith boundary point in such a way that all the networks in the function other than N_i vanish at ith boundary point. Further, the function is trained to satisfy all the given conditions and this trained function is the required trajectory. Using this procedure the walking trajectories can be generated offline (for given constraints as given in Section 5.4). The trained trajectory can be modified online according to change in constraints during walking.

6.3.2 Polynomial Approach for Biped's Gait

The joint trajectories for the boundary conditions given in Section 5.4 are generated using polynomial approach (see Section 4.3.1).

6.4 Simulation Results

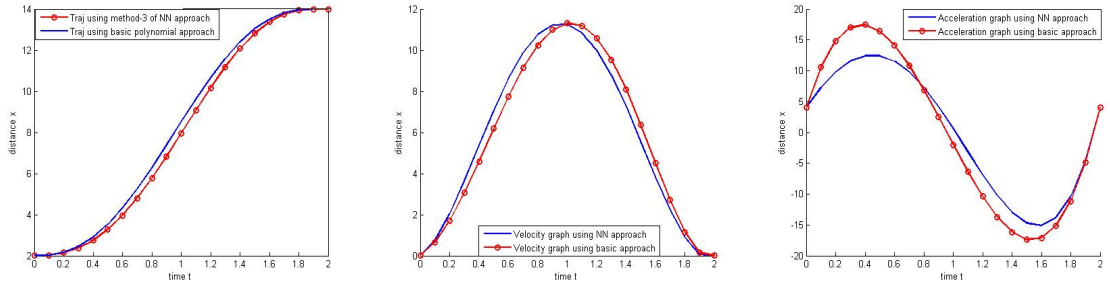
In this simulation study, learning rates α_1 and α_2 in the weight updation algorithm are taken to be 0.0001 and the weights are initialized to be random numbers between 0 and 1.



(a) Position trajectories (b) Velocity trajectories (c) Acceleration trajectories

Figure 6.2: x-t trajectories using Proposed Method versus standard polynomial approach for 4 constraints (x_0=2, x_p=14, x_{v_1}=0, x_{v_2}=0)

The results of the proposed FNN method are compared with the standard polynomial approach for trajectory generation. In the Figures 6.2 and 6.3, there are two trajectories one is generated by using the proposed FNN method (red curve) and other is generated by using standard polynomial approach (blue curve) for 4 and 6 conditions respectively. It can be observed from the Figures 6.2 and 6.3 the trajectories and respective derivative trajectories



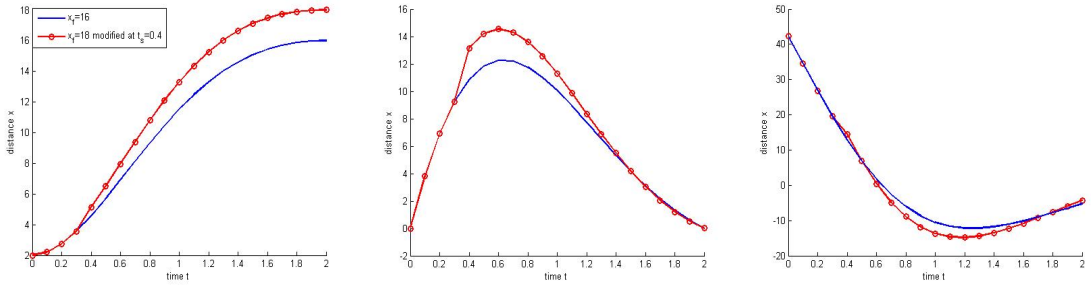
(a) Position trajectories (b) Velocity trajectories (c) Acceleration trajectories

Figure 6.3: x-t trajectories FNN versus standard polynomial approach for 6 constraints (x_0=2, x_p=14, x_{v_1}=0, x_{v_2}=0, $a_{c_1} = 4$, $a_{c_2} = 4$)

are continuous and satisfy the given conditions. In the Figure 6.4(a), the blue curve represent a trajectory using the proposed FNN method for final position x_f=16 while the red curve represents a new trajectory when the final position is changed to x_F=18 at time t_s=0.4 during tracking. Similarly, in the Figure 6.5(a) blue curve represent a trajectory using cubic polynomial for final position x_f=16 while the red curve represents a new trajectory when the final position is changed to x_F=18 at time t_s=0.4 during tracking. Figures 6.4(b), 6.5(b) and 6.4(c), 6.5(c) represents the corresponding velocity and acceleration trajectories respectively (using FNN and WNN). It is worthy to note that the acceleration curve for the modified trajectory generated by proposed FNN method is smooth while the acceleration curve for the changed trajectory generated by the cubic polynomial is discontinuous at time t=0.4.

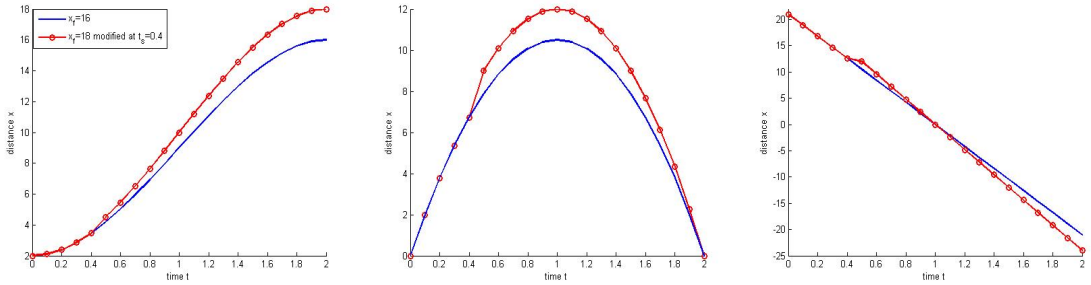
The parameters used for simulation of the trajectories are given in Table 5.5 of Chapter 5. The position, velocity and acceleration trajectories of the upper body using quantic polynomial approach and using proposed NN approach are shown in Figure 6.6(a) and 6.6(b) respectively. Figures 6.7(a) and 6.7(b) demonstrate the ZMP curves for the upper body trajectory using quantic polynomial approach and proposed FNN approach respectively. From the Figures 6.6 and 6.7, it can be concluded that the ZMP curve using proposed FNN is under the supporting polygon while it is outside the region at some time points in case of quantic polynomial.

Torque using PD controller for the polynomial and FNN trajectory for SSP are given in Figures 6.8(a) and 6.8(b) respectively and convergence of error graphs are given in Figures



(a) Position trajectories (b) Velocity trajectories (c) Acceleration trajectories

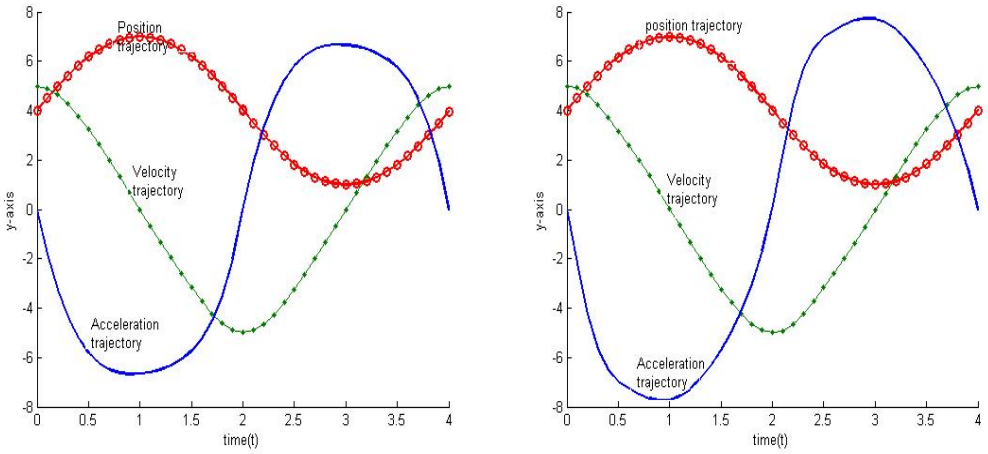
Figure 6.4: If final position changed (using proposed approach)



(a) Position trajectories (b) Velocity trajectories (c) Acceleration trajectories

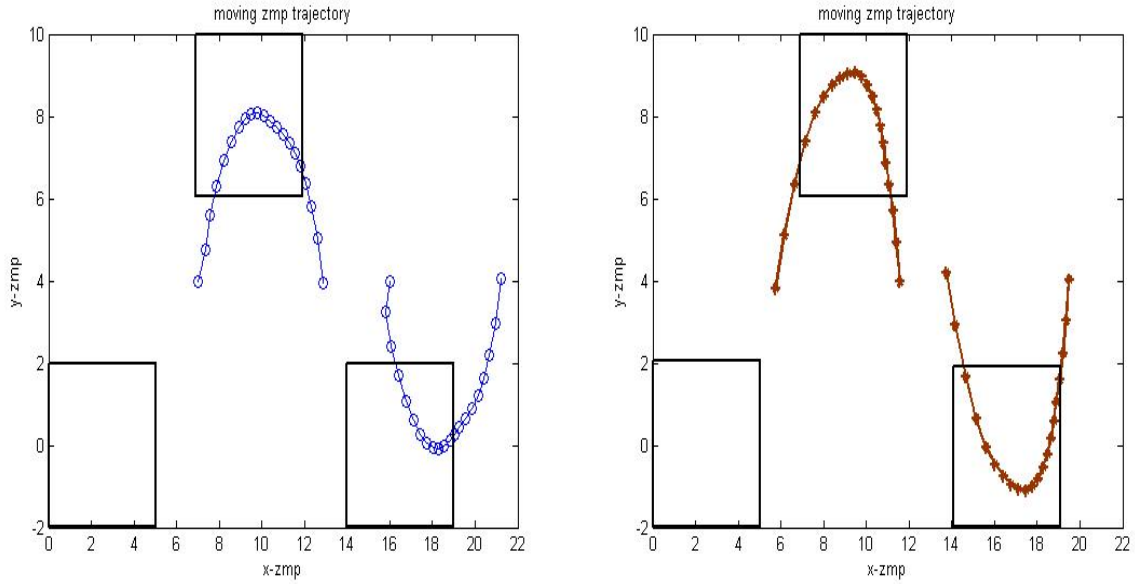
Figure 6.5: If final position changed (using cubic polynomial approach)

6.9(a) and 6.9(b) respectively, which show that NN joint trajectories required less torque in compared to the polynomial joint trajectories. Hence, the FNN method is best suitable for trajectory generation when there are obstacles and ditch in the path of robot and the step length x_f and step height h_1 of ankle trajectory will changed to cross them during tracking. In this simulation, final position is changed from 16 units to 18 units at time t_s=0.3 during the locomotion and the simulation results are shown in Figure 6.10. It took very less time to adapt the change/to train the weights of N_2 because in the learning algorithm the trained weights for x_p=16 are used as initial weights for $x_P = 18$. Similarly, if step height h_1 of ankle trajectory is changed from 2 units to 3 units to cross an obstacle at time t_s=0.4, then ankle trajectory can adopt this change only by updating the weights of N_7 as shown in Figure 6.12. In all these cases, ZMP of the biped is in support region (see Figure 6.11 and 6.13). In 3D space, gait of biped robot for two steps is shown in Figure 6.14.



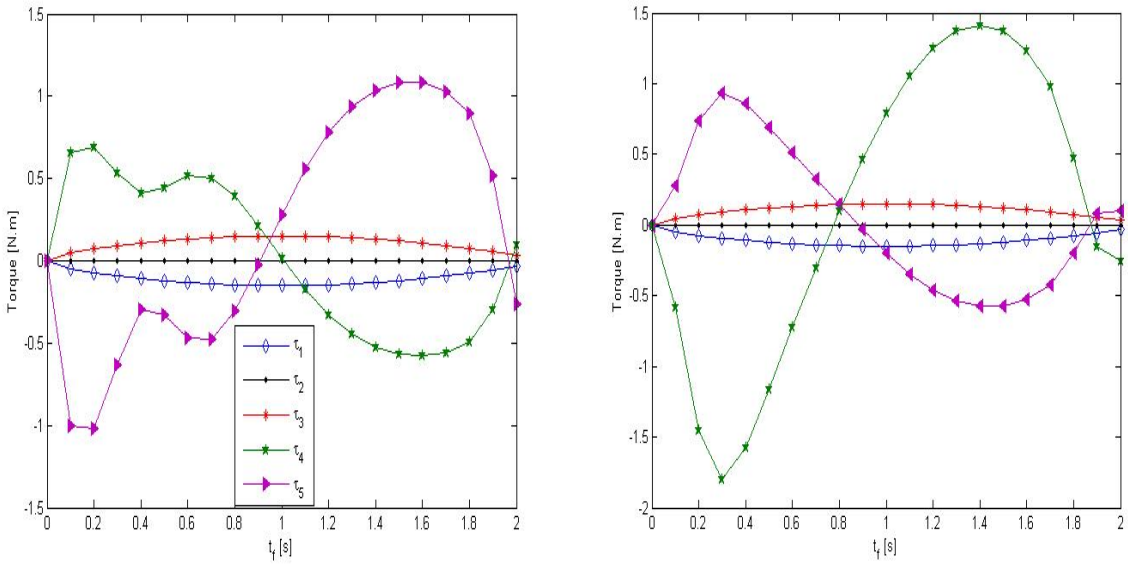
(a) Using polynomial approach (b) Using FNN approach

Figure 6.6: Upper body trajectory in y-direction



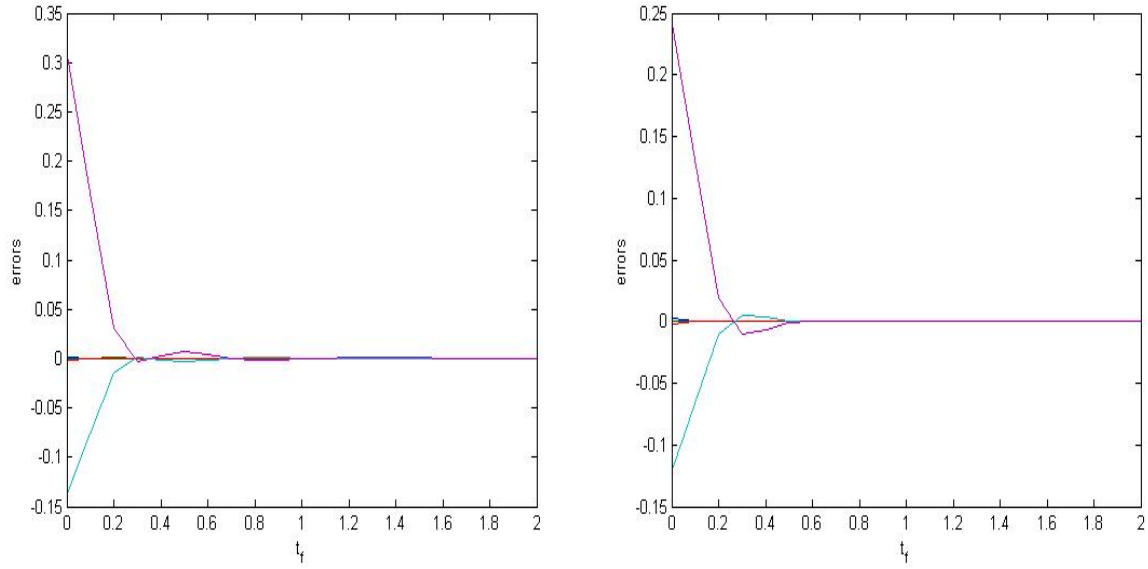
(a) For polynomial upper body trajectory (b) For FNN upper body trajectory

Figure 6.7: ZMP trajectory



(a) For FNN trajectory (b) For polynomial trajectory

Figure 6.8: Required torque using PD controller for SSP



(a) For FNN trajectory (b) For polynomial trajectory

Figure 6.9: Error convergence using PD controller

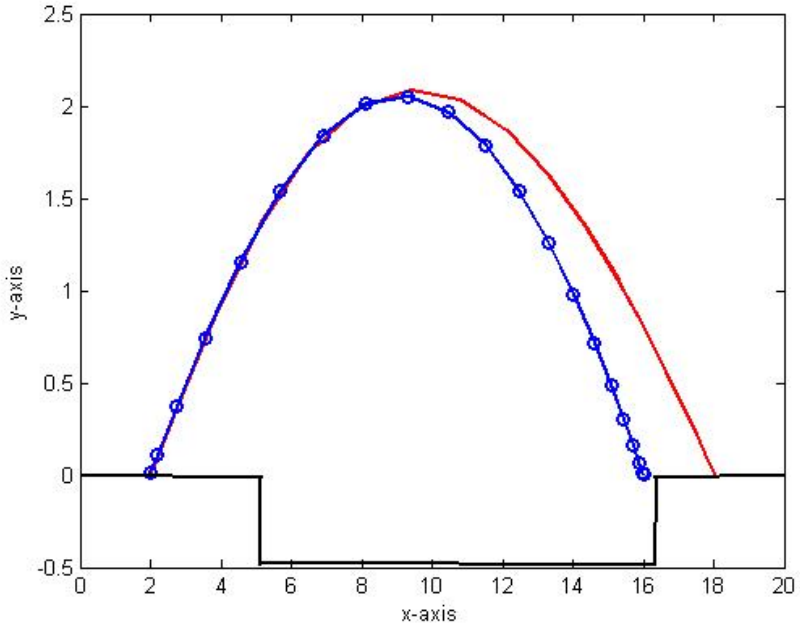


Figure 6.10: Change in ankle trajectory to cross a ditch

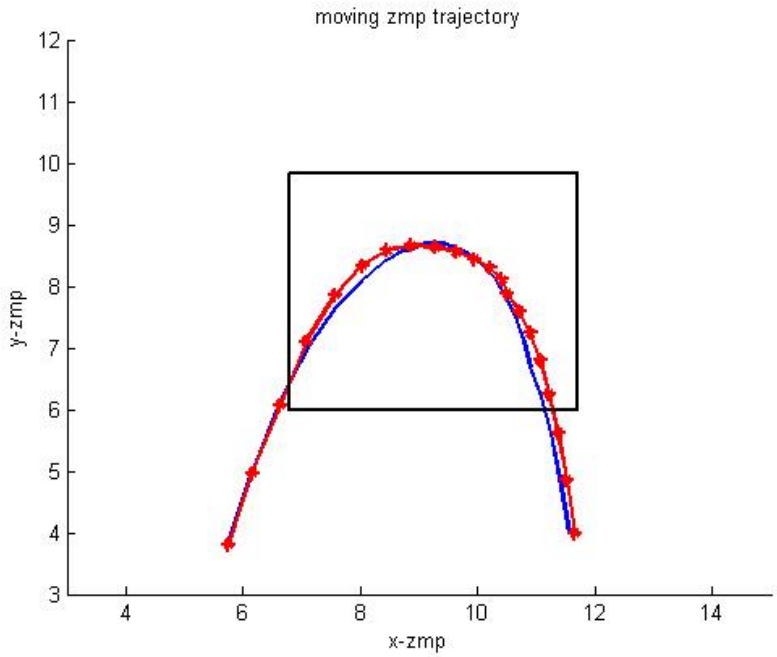


Figure 6.11: ZMP for changed trajectory

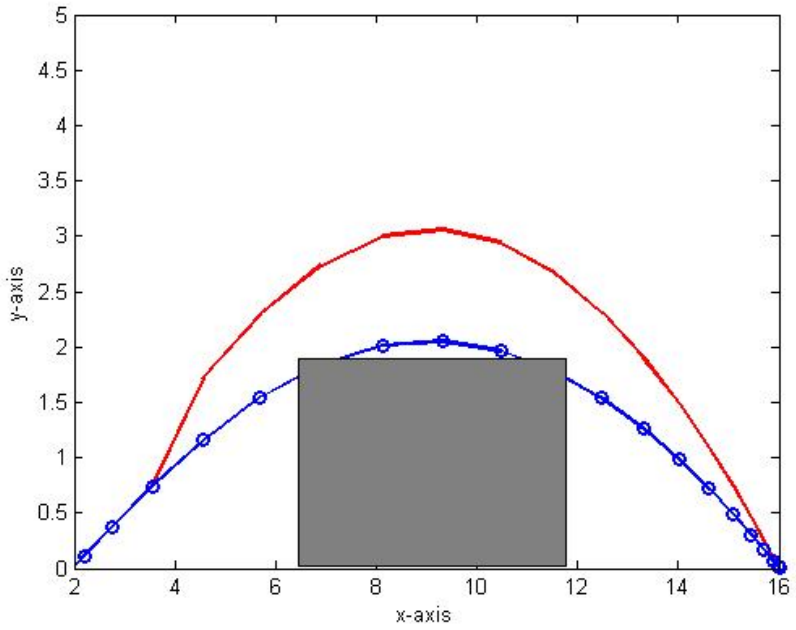


Figure 6.12: Change in ankle trajectory to cross an obstacle

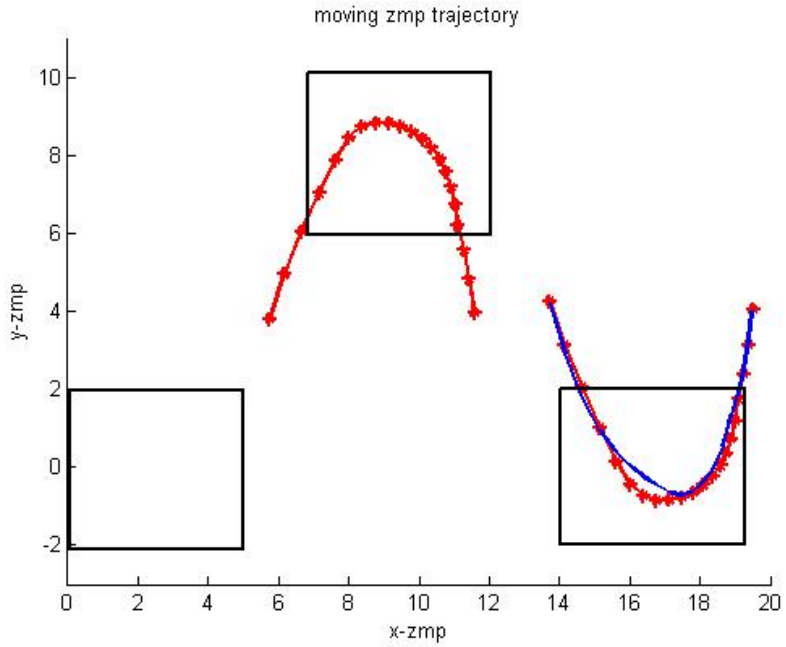


Figure 6.13: ZMP for changed trajectory

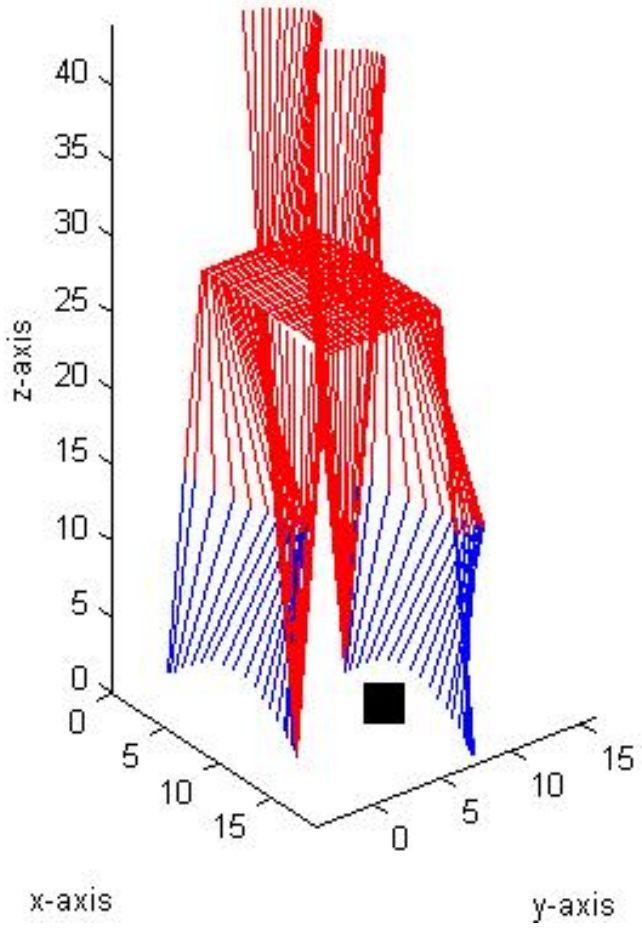


Figure 6.14: Biped walk in 3D of two step for NN trajectories

6.5 Conclusion

For small foot length (5 units), the ZMP curve generated using FNN trajectory is stable while in case of polynomial approach it is unstable (see figures 6.7(a) and 6.7(b)). Successful obstacles negotiation needs online ankle trajectory regulation during tracking. Although the proposed ankle trajectory is changed to cross a ditch or obstacle during walk, the ZMP trajectory is inside the support polygon for these changed trajectories. So, this robot can cross over obstacles of different heights and cross over a ditch by changing the step height and step length in ankle trajectory during tracking at any instant of time.

$$
\begin{aligned}
D_{11} &= (m_2(3l_1^2 + 3\cos(\theta_2)l_1l_2 + l_2^2))/3 + (m_4(3l_1^2 + 6\cos(\theta_2)l_1l_2 + 6\cos(\theta_2+\theta_3+\theta_4)l_1l_2 \\
&+ 3\cos(\theta_2+\theta_3+\theta_4+\theta_5)l_1l_1 + 3l_2^2 + 6\cos(\theta_3+\theta_4)l_2l_2 + 3\cos(\theta_2+\theta_3+\theta_4)l_2l_1 + 3l_2^2 \\
&+ 3\cos(\theta_5)l_2l_1 + l_1^2))/3 + (m_3(3l_1^2 + 6\cos(\theta_2)l_1l_2 + 3\cos(\theta_2+\theta_3+\theta_4)l_1l_2 + 3l_2^2 \\
&+ 3\cos(\theta_3+\theta_4)l_2l_2 + l_2^2))/3 + (m_0+m_5)(l_1^2 + 2\cos(\theta_2)l_1l_2 + l_2^2) + (l_1^2m_1)/3;
\end{aligned}
$$

$$
\begin{aligned}
D_{21} &= (m_4(6l_2^2 + 1_2\cos(\theta_3+\theta_4)l_2l_2 + 6\cos(\theta_3+\theta_4+\theta_5)l_2l_1 + 6l_1\cos(\theta_2)l_2 + 6l_2^2 \\
&+ 6\cos(\theta_5)l_2l_1 + 6l_1\cos(\theta_2+\theta_3+\theta_4)l_2 + 2l_1^2 + 3l_1\cos(\theta_2+\theta_3+\theta_4+\theta_5)l_1))/6 \\
&+ (m_3(6l_2^2 + 6\cos(\theta_3+\theta_4)l_2l_2 + 6l_1\cos(\theta_2)l_2 \\
&+ 2l_2^2 + 3l_1\cos(\theta_2+\theta_3+\theta_4)l_2))/6 - (l_1^2m_1)/6 \\
&+ l_2(m_0+m_5)(l_2 + l_1\cos(\theta_2)) + (l_2m_2(2l_2 + 3l_1\cos(\theta_2)))/6;
\end{aligned}
$$

$$
\begin{aligned}
D_{31} &= (m_4(6l_2^2 + 2l_1^2 + 6l_2l_1\cos(\theta_5) + 6l_1l_2\cos(\theta_2+\theta_3+\theta_4) + 3l_2l_1\cos(\theta_3+\theta_4+\theta_5) \\
&+ 3l_1l_1\cos(\theta_2+\theta_3+\theta_4+\theta_5) + 6l_2l_2\cos(\theta_3+\theta_4)))/6 + (l_2m_3(2l_2 + 3l_2\cos(\theta_3+\theta_4) \\
&+ 3l_1\cos(\theta_2+\theta_3+\theta_4)))/6 - (l_2m_2(l_2 + 3l_1\cos(\theta_2)))/6;
\end{aligned}
$$

$$
\begin{aligned}
D_{41} &= (m_4(6l_2^2 + 2l_1^2 + 6l_2l_1\cos(\theta_5) + 6l_1l_2\cos(\theta_2+\theta_3+\theta_4) + 3l_2l_1\cos(\theta_3+\theta_4+\theta_5) \\
&+ 3l_1l_1\cos(\theta_2+\theta_3+\theta_4+\theta_5) + 6l_2l_2\cos(\theta_3+\theta_4)))/6 + (l_2m_3(2l_2 + 3l_2\cos(\theta_3+\theta_4) \\
&+ 3l_1\cos(\theta_2+\theta_3+\theta_4)))/6;
\end{aligned}
$$

$$
D_{51} = (l_1m_4(2l_1 + 3l_1\cos(\theta_2+\theta_3+\theta_4+\theta_5) + 3l_2\cos(\theta_5) + 3l_2\cos(\theta_3+\theta_4+\theta_5)))/6;
$$

$$
\begin{aligned}
D_{22} &= (m_4(3l_2^2 + 6\cos(\theta_3+\theta_4)l_2l_2 + 3\cos(\theta_3+\theta_4+\theta_5)l_2l_1 + 3l_2^2 + 3\cos(\theta_5)l_2l_1 + l_1^2))/3 \\
&+ l_2^2(m_0+m_5) + (l_1^2m_1)/3 + (l_2^2m_2)/3 + (m_3(3l_2^2 + 3\cos(\theta_3+\theta_4)l_2l_2 + l_2^2))/3;
\end{aligned}
$$

$$
\begin{aligned}
D_{32} &= (m_4(6l_2^2 + 6\cos(\theta_5)l_2l_1 + 6l_2\cos(\theta_3+\theta_4)l_2 + 2l_1^2 + 3l_2\cos(\theta_3+\theta_4+\theta_5)l_1))/6 \\
&- (l_2^2m_2)/6 + (l_2m_3(2l_2 + 3l_2\cos(\theta_3+\theta_4)))/6;
\end{aligned}
$$

$$
\begin{aligned}
D_{42} &= (m_4(6l_2^2 + 6\cos(\theta_5)l_2l_1 + 6l_2\cos(\theta_3+\theta_4)l_2 + 2l_1^2 \\
&+ 3l_2\cos(\theta_3+\theta_4+\theta_5)l_1))/6 + (l_2m_3(2l_2 + 3l_2\cos(\theta_3+\theta_4)))/6;
\end{aligned}
$$

$$
D_{52} = (l_1m_4(2l_1 + 3l_2\cos(\theta_5) + 3l_2\cos(\theta_3+\theta_4+\theta_5)))/6;
$$

$$
D_{33} = (m_4(3l_2^2 + 3\cos(\theta_5)l_2l_1 + l_1^2))/3 + (l_2^2m_2)/3 + (l_2^2m_3)/3;
$$

$$
D_{43} = (m_4(3l_2^2 + 3\cos(\theta_5)l_2l_1 + l_1^2))/3 + (l_2^2m_3)/3;
$$

$$
\begin{aligned}
D_{53} &= (l_1 m_4(2l_1 + 3l_2\cos(\theta_5)))/6;\\
D_{44} &= (m_4(3l_2^2 + 3\cos(\theta_5)l_2 l_1 + l_1^2))/3 + (l_2^2 m_3)/3;\\
D_{54} &= (l_1 m_4(2l_1 + 3l_2\cos(\theta_5)))/6;\\
D_{55} &= (l_1^2 m_4)/3;
\end{aligned}
$$

$$
\begin{aligned}
C_1 &= -\dot{\theta}_1(\dot{\theta}_3((l_2 m_3(l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3\\
&+ \theta_4+\theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4) - (l_1 l_2 m_2\sin(\theta_2))/2 + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4)\\
&+ (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_2((l_1 m_3(2l_2\sin(\theta_2) + l_2\sin(\theta_2+\theta_3+\theta_4)))/2\\
&+ (l_1 m_4(l_1\sin(\theta_2+\theta_3+\theta_4+\theta_5) + 2l_2\sin(\theta_2) + 2l_2\sin(\theta_2+\theta_3+\theta_4)))/2\\
&+ l_1 l_2\sin(\theta_2)(m_0+m_5) + (l_1 l_2 m_2\sin(\theta_2))/2) + \dot{\theta}_4((l_2 m_3(l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4)\\
&+ l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + (\dot{\theta}_5 l_1 m_4(l_1\sin(\theta_2+\theta_3\\
&+ \theta_4+\theta_5) + l_2\sin(\theta_5) + l_2\sin(\theta_3+\theta_4+\theta_5)))/2) - \dot{\theta}_3(\dot{\theta}_1((l_2 m_3(l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4)\\
&- (l_1 l_2 m_2\sin(\theta_2))/2 + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2)\\
&+ \dot{\theta}_2((l_2 m_3(l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2\\
&+ l_2 l_2 m_4\sin(\theta_3+\theta_4) - (l_1 l_2 m_2\sin(\theta_2))/2 + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4)\\
&+ (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_3((l_2 m_3(l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_2+\theta_3+\theta_4)))/2\\
&+ (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4) - (l_1 l_2 m_2\sin(\theta_2))/2\\
&+ l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_4((l_2 m_3(l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4)\\
&+ l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + (\dot{\theta}_5 l_1 m_4(l_1\sin(\theta_2\\
&+ \theta_3+\theta_4+\theta_5) + l_2\sin(\theta_5) + l_2\sin(\theta_3+\theta_4+\theta_5)))/2) - \dot{\theta}_2(\dot{\theta}_3((l_2 m_3(l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2\\
&+ l_2 l_2 m_4\sin(\theta_3+\theta_4) - (l_1 l_2 m_2\sin(\theta_2))/2 + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4)\\
&+ (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_1((l_1 m_3(2l_2\sin(\theta_2) + l_2\sin(\theta_2+\theta_3+\theta_4)))/2\\
&+ (l_1 m_4(l_1\sin(\theta_2+\theta_3+\theta_4+\theta_5) + 2l_2\sin(\theta_2) + 2l_2\sin(\theta_2+\theta_3+\theta_4)))/2 + l_1 l_2\sin(\theta_2)\\
&(m_0+m_5) + (l_1 l_2 m_2\sin(\theta_2))/2) + \dot{\theta}_2((l_1 m_3(2l_2\sin(\theta_2) + l_2\sin(\theta_2+\theta_3+\theta_4)))/2
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad (l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + 2l_2 \sin(\theta_2) + 2l_2 \sin(\theta_2+\theta_3+\theta_4)))/2 + l_1 l_2 \sin(\theta_2) \\
&\quad (m_0+m_5) + (l_1 l_2 m_2 \sin(\theta_2))/2) + \dot{\theta}_4((l_2 m_3(l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_2+\theta_3+\theta_4)))/2 \\
&+ \quad (l_1 l_1 m_4 \sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4 \sin(\theta_3+\theta_4) + l_1 l_2 m_4 \sin(\theta_2+\theta_3+\theta_4) \\
&+ \quad (l_2 l_1 m_4 \sin(\theta_3+\theta_4+\theta_5))/2) + (\dot{\theta}_5 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) \\
&+ \quad l_2 \sin(\theta_3+\theta_4+\theta_5)))/2) - \dot{\theta}_5((\dot{\theta}_1 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) + l_2 \sin(\theta_3 \\
&+ \quad \theta_4+\theta_5)))/2 + (\dot{\theta}_2 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 \\
&+ \quad (\dot{\theta}_3 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_4 l_1 m_4(l_1 \sin(\theta_2 \\
&+ \quad \theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_5 l_1 m_4(l_1 \sin(\theta_2 \\
&+ \quad \theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2) - \dot{\theta}_4(\dot{\theta}_1((l_2 m_3(l_2 \sin(\theta_3+\theta_4) \\
&+ \quad l_1 \sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4 \sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4 \sin(\theta_3+\theta_4) \\
&+ \quad l_1 l_2 m_4 \sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4 \sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_2((l_2 m_3(l_2 \sin(\theta_3+\theta_4) \\
&+ \quad l_1 \sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4 \sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 l_2 l_2 m_4 \sin(\theta_3+\theta_4) + l_1 l_2 m_4 \sin(\theta_2 \\
&+ \quad \theta_3+\theta_4) + (l_2 l_1 m_4 \sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_3((l_2 m_3(l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_2+\theta_3+\theta_4)))/2 \\
&+ \quad (l_1 l_1 m_4 \sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4 \sin(\theta_3+\theta_4) + l_1 l_2 m_4 \sin(\theta_2+\theta_3+\theta_4) \\
&+ \quad (l_2 l_1 m_4 \sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_4((l_2 m_3(l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_2+\theta_3+\theta_4)))/2 \\
&+ \quad (l_1 l_1 m_4 \sin(\theta_2+\theta_3+\theta_4+\theta_5))/2 + l_2 l_2 m_4 \sin(\theta_3+\theta_4) + l_1 l_2 m_4 \sin(\theta_2+\theta_3+\theta_4) \\
&+ \quad (l_2 l_1 m_4 \sin(\theta_3+\theta_4+\theta_5))/2) + (\dot{\theta}_5 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) + l_2 \sin(\theta_5) \\
&+ \quad l_2 \sin(\theta_3+\theta_4+\theta_5)))/2);
\end{aligned}
$$

$$
\begin{aligned}
C_2 \quad = \quad & -\dot{\theta}_2(\dot{\theta}_3((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2) \\
& + \quad \dot{\theta}_4((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2) \\
& + \quad (\dot{\theta}_5 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2) - \dot{\theta}_5((\dot{\theta}_1 l_1 m_4(l_2 \sin(\theta_5) \\
& + \quad l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_2 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 \\
& + \quad (\dot{\theta}_3 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_4 l_1 m_4(l_2 \sin(\theta_5) \\
& + \quad l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_5 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2) \\
& - \quad \dot{\theta}_1(\dot{\theta}_3((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2) \\
& + \quad \dot{\theta}_4((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2)
\end{aligned}
$$

$$
\begin{aligned}
&- \dot{\theta}_1((l_1 m_3(2l_2\sin(\theta_2) + l_2\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 m_4(l_1\sin(\theta_2+\theta_3+\theta_4+\theta_5)\\
&+ 2l_2\sin(\theta_2) + 2l_2\sin(\theta_2+\theta_3+\theta_4)))/2 + l_1 l_2\sin(\theta_2)(m_0+m_5)\\
&+ (l_1 l_2 m_2\sin(\theta_2))/2) + (\dot{\theta}_5 l_1 m_4(l_2\sin(\theta_5) + l_2\sin(\theta_3+\theta_4+\theta_5)))/2)\\
&- \dot{\theta}_3(\dot{\theta}_1((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_2((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_3((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_4((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ (\dot{\theta}_5 l_1 m_4(l_2\sin(\theta_5) + l_2\sin(\theta_3+\theta_4+\theta_5)))/2)\\
&- \dot{\theta}_4(\dot{\theta}_1((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_2((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_3((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_4((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ (\dot{\theta}_5 l_1 m_4(l_2\sin(\theta_5) + l_2\sin(\theta_3+\theta_4+\theta_5)))/2);
\end{aligned}
$$

$$
\begin{aligned}
C_3 \;=\;& \dot{\theta}_1(\dot{\theta}_1((l_2 m_3(l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4\\
&+ \theta_5))/2 + l_2 l_2 m_4\sin(\theta_3+\theta_4) - (l_1 l_2 m_2\sin(\theta_2))/2 + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4)\\
&+ (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2) + \dot{\theta}_2((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2\\
&+ (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2) - (\dot{\theta}_5 l_2 l_1 m_4\sin(\theta_5))/2) + \dot{\theta}_2(\dot{\theta}_1((l_2 m_4(2l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2) + \dot{\theta}_2((l_2 m_4(2l_2\sin(\theta_3+\theta_4)\\
&+ l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2) - (\dot{\theta}_5 l_2 l_1 m_4\sin(\theta_5))/2)\\
&- \dot{\theta}_5((\dot{\theta}_1 l_2 l_1 m_4\sin(\theta_5))/2 + (\dot{\theta}_2 l_2 l_1 m_4\sin(\theta_5))/2 + (\dot{\theta}_3 l_2 l_1 m_4\sin(\theta_5))/2\\
&+ (\dot{\theta}_4 l_2 l_1 m_4\sin(\theta_5))/2 + (\dot{\theta}_5 l_2 l_1 m_4\sin(\theta_5))/2) - (\dot{\theta}_3\dot{\theta}_5 l_2 l_1 m_4\sin(\theta_5))/2\\
&- (\dot{\theta}_4\dot{\theta}_5 l_2 l_1 m_4\sin(\theta_5))/2;
\end{aligned}
$$

$$
\begin{aligned}
C_4 \;=\;& \dot{\theta}_1(\dot{\theta}_2((l_2 m_4(2l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_3+\theta_4+\theta_5)))/2 + (l_2 l_2 m_3\sin(\theta_3+\theta_4))/2)\\
&+ \dot{\theta}_1((l_2 m_3(l_2\sin(\theta_3+\theta_4) + l_1\sin(\theta_2+\theta_3+\theta_4)))/2 + (l_1 l_1 m_4\sin(\theta_2+\theta_3+\theta_4+\theta_5))/2\\
&+ l_2 l_2 m_4\sin(\theta_3+\theta_4) + l_1 l_2 m_4\sin(\theta_2+\theta_3+\theta_4) + (l_2 l_1 m_4\sin(\theta_3+\theta_4+\theta_5))/2)
\end{aligned}
$$

$$
\begin{aligned}
&- (\dot{\theta}_5 l_2 l_1 m_4 \sin(\theta_5))/2) + \dot{\theta}_2(\dot{\theta}_1((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 \\
&+ (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2) + \dot{\theta}_2((l_2 m_4(2l_2 \sin(\theta_3+\theta_4) + l_1 \sin(\theta_3+\theta_4+\theta_5)))/2 \\
&+ (l_2 l_2 m_3 \sin(\theta_3+\theta_4))/2) - (\dot{\theta}_5 l_2 l_1 m_4 \sin(\theta_5))/2) - \dot{\theta}_5((\dot{\theta}_1 l_2 l_1 m_4 \sin(\theta_5))/2 \\
&+ (\dot{\theta}_2 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_3 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_4 l_2 l_1 m_4 \sin(\theta_5))/2 \\
&+ (\dot{\theta}_5 l_2 l_1 m_4 \sin(\theta_5))/2) - (\dot{\theta}_3 \dot{\theta}_5 l_2 l_1 m_4 \sin(\theta_5))/2 - (\dot{\theta}_4 \dot{\theta}_5 l_2 l_1 m_4 \sin(\theta_5))/2;
\end{aligned}
$$

$$
\begin{aligned}
C_5 &= \dot{\theta}_1((\dot{\theta}_2 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_1 l_1 m_4(l_1 \sin(\theta_2+\theta_3+\theta_4+\theta_5) \\
&+ l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_3 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_4 l_2 l_1 m_4 \sin(\theta_5))/2) \\
&+ \dot{\theta}_3((\dot{\theta}_1 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_2 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_3 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_4 l_2 l_1 m_4 \sin(\theta_5))/2) \\
&+ \dot{\theta}_4((\dot{\theta}_1 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_2 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_3 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_4 l_2 l_1 m_4 \sin(\theta_5))/2) \\
&+ \dot{\theta}_2((\dot{\theta}_1 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 + (\dot{\theta}_2 l_1 m_4(l_2 \sin(\theta_5) + l_2 \sin(\theta_3+\theta_4+\theta_5)))/2 \\
&+ (\dot{\theta}_3 l_2 l_1 m_4 \sin(\theta_5))/2 + (\dot{\theta}_4 l_2 l_1 m_4 \sin(\theta_5))/2);
\end{aligned}
$$

$$
\begin{aligned}
G_1 &= (g l_2 m_3 \cos(\theta_1+\theta_2+\theta_3+\theta_4))/2 - g(m_0+m_5)(l_2 \cos(\theta_1+\theta_2) + l_1 \cos(\theta_1)) \\
&- g m_2(l_2 \cos(\theta_1+\theta_2) + l_1 \cos(\theta_1)) - g m_3(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_2 \cos(\theta_1 \\
&+ \theta_2) + l_1 \cos(\theta_1)) - g m_4(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_2 \cos(\theta_1+\theta_2) + l_1 \cos(\theta_1 \\
&+ \theta_2+\theta_3+\theta_4+\theta_5) + l_1 \cos(\theta_1)) + (g l_2 m_2 \cos(\theta_1+\theta_2))/2 \\
&+ (g l_1 m_4 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5))/2 - (g l_1 m_1 \cos(\theta_1))/2; \\
G_2 &= (g l_2 m_3 \cos(\theta_1+\theta_2+\theta_3+\theta_4))/2 - g m_4(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_2 \cos(\theta_1+\theta_2) \\
&+ l_1 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5)) - g m_3(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_2 \cos(\theta_1+\theta_2)) \\
&- (g l_2 m_2 \cos(\theta_1+\theta_2))/2 + (g l_1 m_4 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5))/2 \\
&- g l_2 \cos(\theta_1+\theta_2)(m_0+m_5); \\
G_3 &= (g l_1 m_4 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5))/2 - (g l_2 m_3 \cos(\theta_1+\theta_2+\theta_3+\theta_4))/2 \\
&- g m_4(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_1 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5)); \\
G_4 &= (g l_1 m_4 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5))/2 - (g l_2 m_3 \cos(\theta_1+\theta_2+\theta_3+\theta_4))/2 \\
&- g m_4(l_2 \cos(\theta_1+\theta_2+\theta_3+\theta_4) + l_1 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5)); \\
G_5 &= -(g l_1 m_4 \cos(\theta_1+\theta_2+\theta_3+\theta_4+\theta_5))/2;
\end{aligned}
$$

Chapter 7: Conclusions and Future Scope

7.1 Conclusions

In this thesis, some flat footed and toe footed biped robot models are considered for stable human like gait while walking on plane and uneven surfaces. The joint trajectories are designed such that the ZMP lies in the support region with largest stability margin. These trajectories can be modified according to the obstacles and ditch in the path. In the proposed walking patterns, the robot can walk at low and medium walking speeds. In the following lines, chapterwise conclusions are discussed in brief:

In Chapter 3, trajectories are generated by using polynomial approach for upper body, ankle and hip joints of flat footed robot. Three types of upper body trajectories are generated and their effects are analyzed on the ZMP stability with suitable boundary conditions. Simulation results show that Case-3 of upper body trajectory ensures the ZMP stability with largest stability margin. Inverse kinematics of these trajectories are calculated by using FNN.

In Chapter 4, three types of ankle trajectories are considered for smooth transition between the walking phases(SSP to DSP) along with three cases of lateral upper body motion, to analyze their effect on ZMP stability. Movements at ankle and hip joints are added to realize human like walk. The simulation results show that Case-1 of upper body motion with quantic polynomial ankle trajectory with optimum parameters is more suitable for ZMP stability with largest stability margin. So it can be concluded that upper body mass should be shifted from swing foot to stable foot during DSP for the proposed model. Simulation results show that this model can walk on plane and uneven surfaces.

In Chapter 5, FNN and WNN approaches are proposed for trajectory generation and have been illustrated for 4 and 6 constraints was carried out using Matlab2014b results were

compared using Matlab2014b. The trajectories generated by using proposed approaches can be modified at any instant during tracking. Further, these approaches are used to generate trajectories of the upper body, foot and hip for human like walk of flat footed biped robot model. The simulated results show that WNN has 3 times faster convergence in comparison with FNN. However, FNN generates more smooth trajectories which is required for stable biped robot walk. However, WNN approach can be applied to generate ankle trajectory for a robot model with increased foot size as stability is not a big issue in that case.

In Chapter 6, the dynamic equation is derived for a 5 DOF flat footed biped robot model. The FNN method proposed in Chapter 5 is adopted for gait generation and results are compared with the standard polynomial approach. FNN approach is more suitable for small foot length as compared to polynomial approach. Furthermore, this approach can adapt the changes in constraints according to uncertain environment/terrain (obstacles and ditch) and modify the trajectory accordingly during tracking. The ZMP is inside the support polygon for modified ankle trajectories also. Hence, this robot can cross over the obstacles and ditch during tracking by changing the step length and step height.

7.2 Future Scope

Although the proposed models can perform walking satisfactorily, there are some limitation in this work. However, to walk on stairs, decline surface, step over an large obstacles, biped models with higher DOF are essential. The proposed FNN and WNN approaches can be implemented on the higher degree models for more adaptable walk on different surfaces. Further, the PD controller is used to control the proposed gaits for biped robot whose efficiency decreases when there is a disturbance, so suitable controller can be developed and energy optimization can be considered.

Bibliography

[1] Abe, A. (2011). Trajectory planning for flexible cartesian robot manipulator by using artificial neural network: numerical simulation and experimental verification. *Robotica*, 29(5):797–804.

[2] Aghaabbasloo, M., Azarkaman, M., and Salehi, M. E. (2013). Biped robot joint trajectory generation using pso evolutionary algorithm. In *2013 3rd Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*, pages 1–6. IEEE.

[3] Akilandeswari, J. and Gopalan, N. (2005). A web mining system using reinforcement learning for scalable web search with distributed. fault-tolerant multi-agents. *WSEAS transactions on Computers*, 4:1633–1640.

[4] Al-Abdullah, K. I., Lim, C. P., Najdovski, Z., and Yassin, W. (2019). A model-based bone milling state identification method via force sensing for a robotic surgical system. *The International Journal of Medical Robotics and Computer Assisted Surgery*, page e1989.

[5] Almusawi, A. R., Dülger, L. C., and Kapucu, S. (2016). A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242). *Computational intelligence and neuroscience*, 2016.

[6] Barghi Jond, H., V Nabiyev, V., and Benveniste, R. (2016). Trajectory planning using high order polynomials under acceleration constraint. *Journal of Optimization in Industrial Engineering*, 10(21):1–6.

[7] Behera, L., Chaudhury, S., and Gopal, M. (1996). Neuro-adaptive hybrid controller for

robot-manipulator tracking control. *IEE Proceedings-Control Theory and Applications*, 143(3):270–275.

[8] Bhatnagar, G. and Wu, Q. J. (2012). An image fusion framework based on human visual system in framelet domain. *International Journal of Wavelets, Multiresolution and Information Processing*, 10(01):1250002.

[9] Bhatnagar, G., Wu, Q. J., and Liu, Z. (2013). Human visual system inspired multimodal medical image fusion framework. *Expert Systems with Applications*, 40(5):1708–1720.

[10] Boryga, M. and Graboś, A. (2009). Planning of manipulator motion trajectory with higher-degree polynomials use. *Mechanism and machine theory*, 44(7):1400–1419.

[11] Branner, B. and Hjorth, P. (2013). *Real and complex dynamical systems*, volume 464. Springer Science & Business Media.

[12] Capi, G., Nasu, Y., Barolli, L., Mitobe, K., and Takeda, K. (2001). Application of genetic algorithms for biped robot gait synthesis optimization during walking and going up-stairs. *Advanced robotics*, 15(6):675–694.

[13] Capi, G., Nasu, Y., Mitobe, K., and Barolli, L. (2002). Autonomous humanoid robot locomotion based on neural networks. *Industrial Robot: An International Journal*, 29(3):252–258.

[14] Chaudhary, H., Panwar, V., Prasad, R., and Sukavanam, N. (2016). Adaptive neuro fuzzy based hybrid force/position control for an industrial robot manipulator. *Journal of Intelligent Manufacturing*, 27(6):1299–1308.

[15] Cuevas, E., Zaldivar, D., Perez, M., and Ramirez, M. (2014). Polynomial trajectory algorithm for a biped robot. *arXiv preprint arXiv:1405.5937*.

[16] De Lope, J., González-Careaga, R., Zarraonandia, T., and Maravall, D. (2003). Inverse kinematics for humanoid robots using artificial neural networks. In *International Conference on Computer Aided Systems Theory*, pages 448–459. Springer.

[17] Duka, A.-V. (2014). Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Procedia Technology*, 12:20–27.

[18] Erbatur, K. and Kurt, O. (2006). Humanoid walking robot control with natural zmp references. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, pages 4100–4106. IEEE.

[19] Ferrari, S. and Stengel, R. F. (2005). Smooth function approximation using neural networks. *IEEE Transactions on Neural Networks*, 16(1):24–38.

[20] Fu, K. S., Gonzalez, R., and Lee, C. G. (1987). *Robotics: Control Sensing. Vis.* Tata McGraw-Hill Education.

[21] Fujita, M. (2000). Digital creatures for future entertainment robotics. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 801–806. IEEE.

[22] Gao, W., Jia, Z., and Fu, C. (2017). Increase the feasible step region of biped robots through active vertical flexion and extension motions. *Robotica*, 35(7):1541–1561.

[23] Gienger, M., Loffler, K., and Pfeiffer, F. (2001). Towards the design of a biped jogging robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 4140–4145. IEEE.

[24] Goswami, A. (1999). Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6):523–533.

[25] Gouaillier, D., Collette, C., and Kilner, C. (2010). Omni-directional closed-loop walk for nao. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 448–454. IEEE.

[26] Gupta, G. and Dutta, A. (2018). Trajectory generation and step planning of a 12 dof biped robot on uneven surface. *Robotica*, 36(7):945–970.

[27] Hayat, A. A., Boby, R. A., and Saha, S. K. (2019). A geometric approach for kinematic identification of an industrial robot using a monocular camera. *Robotics and Computer-Integrated Manufacturing*, 57:329–346.

[28] Herzog, S., Wörgötter, F., and Kulvicius, T. (2017). Generation of movements with boundary conditions based on optimal control theory. *Robotics and Autonomous Systems*, 94:1–11.

[29] Hong, S., Oh, Y., Kim, D., and You, B.-J. (2014). Real-time walking pattern generation method for humanoid robots by combining feedback and feedforward controller. *IEEE transactions on industrial electronics*, 61(1):355–364.

[30] Huan, T. T., Van Kien, C., Anh, H. P. H., and Nam, N. T. (2018). Adaptive gait generation for humanoid robot using evolutionary neural model optimized with modified differential evolution technique. *Neurocomputing*.

[31] Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., and Tanie, K. (2001). Planning walking patterns for a biped robot. *IEEE Transactions on robotics and automation*, 17(3):280–289.

[32] Husty, M. L., Pfurner, M., and Schröcker, H.-P. (2007). A new and efficient algorithm for the inverse kinematics of a general serial 6r manipulator. *Mechanism and machine theory*, 42(1):66–81.

[33] Ishida, T. (2004). Development of a small biped entertainment robot qrio. In *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.*, pages 23–28. IEEE.

[34] Janardhan, V. and Kumar, R. P. (2017). Online trajectory generation for wide ditch crossing of biped robots using control constraints. *Robotics and Autonomous Systems*, 97:61–82.

[35] Johnson, M., Shrewsbury, B., Bertrand, S., Wu, T., Duran, D., Floyd, M., Abeles, P., Stephen, D., Mertins, N., Lesman, A., et al. (2015). Team ihmc's lessons learned from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208.

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

[37] Kajita, S., Yamaura, T., and Kobayashi, A. (1992). Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Transactions on robotics and automation*, 8(4):431–438.

[38] Kaneko, K., Kanehiro, F., Kajita, S., Yokoyama, K., Akachi, K., Kawasaki, T., Ota, S., and Isozumi, T. (2002). Design of prototype humanoid robotics platform for hrp. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2431–2436. IEEE.

[39] Kaneko, K., Kanehiro, F., Morisawa, M., Akachi, K., Miyamori, G., Hayashi, A., and Kanehira, N. (2011). Humanoid robot hrp-4-humanoid robotics platform with lightweight and slim body. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4400–4407. IEEE.

[40] Kano, H., Fujioka, H., and Martin, C. F. (2011). Optimal smoothing and interpolating splines with constraints. *Applied Mathematics and Computation*, 218(5):1831–1844.

[41] Kar, I. and Behera, L. (2009). Direct adaptive neural control for affine nonlinear systems. *Applied soft computing*, 9(2):756–764.

[42] Khadiv, M., Moosavian, S. A. A., Yousefi-Koma, A., Maleki, H., and Sadedel, M. (2017). Online adaptation for humanoids walking on uncertain surfaces. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 231(4):245–258.

[43] Kherici, N. and Ali, Y. M. B. (2014). Using pso for a walk of a biped robot. *Journal of Computational Science*, 5(5):743–749.

[44] Khodadadi, H., Razavi, S. E., and Ahmadi-Noubari, H. (2012). A comparison between neural networks and wavelet networks in nonlinear system identification. *Research Journal of Applied Sciences, Engineering and Technology*, 4(9):1021–1026.

[45] Knudsen, J. M. and Hjorth, P. G. (2012). *Elements of Newtonian mechanics: including nonlinear dynamics*. Springer Science & Business Media.

[46] Kumar, N., Panwar, V., Borm, J.-H., Chai, J., and Yoon, J. (2013). Adaptive neural controller for space robot system with an attitude controlled base. *Neural Computing and Applications*, 23(7-8):2333–2340.

[47] Kumar, N., Panwar, V., Sukavanam, N., Sharma, S. P., and Borm, J.-H. (2011). Neural network based hybrid force/position control for robot manipulators. *International Journal of Precision Engineering and Manufacturing*, 12(3):419–426.

[48] Kumar, R. P., Handharu, N., Yoon, J., and Kim, G.-s. (2007). Hybrid toe and heel joints for biped/humanoid robots for natural gait. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 2687–2692. IEEE.

[49] Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000.

[50] Li, Q., Takanishi, A., and Kato, I. (1991). A biped walking robot having a zmp measurement system using universal force-moment sensors. In *Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91*, pages 1568–1573. IEEE.

[51] Li, T., Ceccarelli, M., Luo, M., Laribi, M. A., and Zeghloul, S. (2014). An experimental analysis of overcoming obstacle in human walking. *Journal of Bionic Engineering*, 11(4):497–505.

[52] Lim, C. P. and Harrison, R. F. (1997). An incremental adaptive network for on-line supervised learning and probability estimation. *Neural networks*, 10(5):925–939.

[53] Lim, H.-o., Ogura, Y., and Takanishi, A. (2008). Locomotion pattern generation and mechanisms of a new biped walking machine. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 464, pages 273–288. The Royal Society.

[54] Lim, H.-O. and Takanishi, A. (2006). Biped walking robots created at waseda university: Wl and wabian family. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):49–64.

[55] Lim, H.-o., Yamamoto, Y., and Takanishi, A. (2000). Control to realize human-like walking of a biped humanoid robot. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, volume 5, pages 3271–3276. IEEE.

[56] Liu, C., Wang, D., Goodman, E. D., and Chen, Q. (2016a). Adaptive walking control of biped robots using online trajectory generation method based on neural oscillators. *Journal of Bionic Engineering*, 13(4):572–584.

[57] Liu, G. H., Chen, M. Z., and Chen, Y. (2019). When joggers meet robots: the past, present, and future of research on humanoid robots. *Bio-Design and Manufacturing*, pages 1–11.

[58] Liu, H., Lai, X., and Wu, W. (2013). Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robotics and Computer-Integrated Manufacturing*, 29(2):309–317.

[59] Liu, J. and Urbann, O. (2016). Bipedal walking with dynamic balance that involves three-dimensional upper body motion. *Robotics and Autonomous Systems*, 77:39–54.

[60] Liu, Z., Wang, L., Zhang, Y., and Chen, C. P. (2016b). A svm controller for the stable walking of biped robots based on small sample sizes. *Applied Soft Computing*, 38:738–753.

[61] Martinel, N., Micheloni, C., and Foresti, G. L. (2015). The evolution of neural learning systems: a novel architecture combining the strengths of nts, cnns, and elms. *IEEE Systems, Man, and Cybernetics Magazine*, 1(3):17–26.

[62] Miller, W. T. (1994). Real-time neural network control of a biped walking robot. *IEEE Control Systems Magazine*, 14(1):41–48.

[63] Miura, H. and Shimoyama, I. (1984). Dynamic walk of a biped. *The International Journal of Robotics Research*, 3(2):60–74.

[64] Miyazaki, F. and Arimoto, S. (1980). A control theoretic study on dynamical biped locomotion. *Journal of Dynamic Systems, Measurement, and Control*, 102(4):233–239.

[65] Mousavi, P. N. and Bagheri, A. (2007). Mathematical simulation of a seven link biped robot on various surfaces and zmp considerations. *Applied Mathematical Modelling*, 31(1):18–37.

[66] Muecke, K. and Hong, D. (2011). Constrained analytical trajectory filter for stabilizing humanoid robot motions. *Intelligent Service Robotics*, 4(3):203–218.

[67] Narváez-Aroche, O., Rocha-Cózatl, E., and Cuenca-Jiménez, F. (2011). Kinematic analysis and computation of zmp for a 12-internal-dof biped robot. *Proc. of the 13th World Cong. in Mechanism and Machine Science (IFTOMM)*.

[68] Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., and Inoue, H. (2002). Toe joints that enhance bipedal and fullbody motion of humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 3105–3110. IEEE.

[69] Nori, F., Traversaro, S., Eljaik, J., Romano, F., Del Prete, A., and Pucci, D. (2015). icub whole-body control through force regulation on rigid non-coplanar contacts. *Frontiers in Robotics and AI*, 2:6.

[70] Panwar, V. (2017). Wavelet neural network-based h? trajectory tracking for robot manipulators using fast terminal sliding mode control. *Robotica*, 35(7):1488–1503.

[71] Parisi, D. R., Mariani, M. C., and Laborde, M. A. (2003). Solving differential equations with unsupervised neural networks. *Chemical Engineering and Processing: Process Intensification*, 42(8-9):715–721.

[72] Park, C.-S. and Kim, D. (2013). Classification of obstacle shape for generating walking path of humanoid robot. *Transactions of the Korean Society of Mechanical Engineers A*, 37(2):169–176.

[73] Park, I.-W., Kim, J.-Y., and Oh, J.-H. (2008). Online walking pattern generation and its application to a biped humanoid robotkhr-3 (hubo). *Advanced Robotics*, 22(2-3):159–190.

[74] Park, J. H. (2003). Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy sets and systems*, 134(1):189–203.

[75] Piciarelli, C., Micheloni, C., and Foresti, G. L. (2008). Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for video Technology*, 18(11):1544–1554.

[76] Plestan, F., Grizzle, J. W., Westervelt, E. R., and Abba, G. (2003). Stable walking of a 7-dof biped robot. *IEEE Transactions on Robotics and Automation*, 19(4):653–668.

[77] Rahmatizadeh, R., Abolghasemi, P., and Bölöni, L. (2016). Learning manipulation trajectories using recurrent neural networks. *arXiv preprint arXiv:1603.03833*.

[78] Rubin, S. and Khosla, P. (1979). Numerical methods based on polynomial spline interpolation. In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 370–377. Springer.

[79] Saha, S. K. (2014). *Introduction to robotics*. Tata McGraw-Hill Education.

[80] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., and Fujimura, K. (2002). The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE.

[81] Sarkar, A. and Dutta, A. (2015). 8-dof biped robot with compliant-links. *Robotics and Autonomous Systems*, 63:57–67.

[82] Sarkar, A. and Dutta, A. (2018). Optimal trajectory generation and design of an 8-dof compliant biped robot for walk on inclined ground. *Journal of Intelligent & Robotic Systems*, pages 1–20.

[83] Schilling, R. J. (1990). *Fundamentals of robotics: analysis and control*, volume 629. Prentice Hall New Jersey.

[84] Semwal, V. B., Raj, M., and Nandi, G. C. (2015). Biometric gait identification based on a multilayer perceptron. *Robotics and Autonomous Systems*, 65:65–75.

[85] Shanker, K. S. R., Mala, C., and Gopalan, N. (2008). Human-robot coordination for multi-sized quadrupled robot using wireless communication. In *2008 IEEE International Conference on Technologies for Practical Robot Applications*, pages 116–121. IEEE.

[86] Shaw, A. A. and Gopalan, N. (2014). Finding longest frequent trajectory of dynamic objects using association approaches. *Intelligent Data Analysis*, 18(4):637–651.

[87] Shiller, Z. (2015). Off-line and on-line trajectory planning. In *Motion and Operation Planning of Robotic Systems*, pages 29–62. Springer.

[88] Shin, H.-K. and Kim, B. K. (2014). Energy-efficient gait planning and control for biped robots utilizing the allowable zmp region. *IEEE Transactions on Robotics*, 30(4):986–993.

[89] Singh, H. and Sukavanam, N. (2012). Simulation and stability analysis of neural network based control scheme for switched linear systems. *ISA transactions*, 51(1):105–110.

[90] Stasse, O., Verrelst, B., Vanderborght, B., and Yokoi, K. (2009). Strategies for humanoid robots to dynamically walk over large obstacles. *IEEE Transactions on Robotics*, 25(4):960–967.

[91] Sukaavanam, N. and Panwar, V. (2003). Computation of boundary control of controlled heat equation using artificial neural networks. *International communications in heat and mass transfer*, 30(8):1137–1146.

[92] Takanishi, A., Ishida, M., Yamazaki, Y., and Kato, I. (1985). The realization of dynamic walking by the biped walking robot wl-10 rd. *Journal of the Robotics Society of Japan*, 3(4):325–336.

[93] Tsagarakis, N. G., Caldwell, D. G., Negrello, F., Choi, W., Baccelliere, L., Loc, V.-G., Noorden, J., Muratore, L., Margan, A., Cardellino, A., et al. (2017). Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259.

[94] Vadakkepat, P., Sin, N. B., Goswami, D., Zhang, R. X., and Tan, L. Y. (2009). Soccer playing humanoid robots: Processing architecture, gait generation and vision system. *Robotics and Autonomous Systems*, 57(8):776–785.

[95] Vukobratović, M. and Borovac, B. (2004). Zero-moment pointthirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173.

[96] Vukobratović, M. and Stepanenko, J. (1972). On the stability of anthropomorphic systems. *Mathematical biosciences*, 15(1-2):1–37.

[97] Vundavilli, P. R. and Pratihar, D. K. (2009). Soft computing-based gait planners for a dynamically balanced biped robot negotiating sloping surfaces. *Applied Soft Computing*, 9(1):191–208.

[98] Vundavilli, P. R. and Pratihar, D. K. (2010). Dynamically balanced optimal gaits of a ditch-crossing biped robot. *Robotics and Autonomous Systems*, 58(4):349–361.

[99] Wang, T., Chevallereau, C., and Tlalolini, D. (2014). Stable walking control of a 3d biped robot with foot rotation. *Robotica*, 32(4):551–570.

[100] Xiaoguang, Z. and Ruyi, H. (2015). Research on humanoid robot slope gait planning. *Open Automation and Control Systems Journal*, 7:1002–1009.

[101] Xu, W., Huang, Q., Li, J., Yu, Z., Chen, X., and Xu, Q. (2011). An improved zmp trajectory design for the biped robot bhr. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 569–574. IEEE.

[102] Yoon, J., Novandy, B., Yoon, C.-H., and Park, K.-J. (2010). A 6-dof gait rehabilitation robot with upper and lower limb connections that allows walking velocity updates on various terrains. *IEEE/ASME Transactions on Mechatronics*, 15(2):201–215.

[103] Zheng, Y. F. and Shen, J. (1990). Gait synthesis for the sd-2 biped robot to climb sloping surface. *IEEE Transactions on Robotics and Automation*, 6(1):86–96.

[104] Zhu, H., Luo, M., and Li, J. (2018). Optimization-based gait planning and control for biped robots utilizing the optimal allowable zmp variation region. *Industrial Robot: An International Journal*, 45(4):469–480.

[105] Zhu, H., Luo, M., Mei, T., Zhao, J., Li, T., and Guo, F. (2016). Energy-efficient bio-inspired gait planning and control for biped robot based on human locomotion analysis. *Journal of Bionic Engineering*, 13(2):271–282.