Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)

Tilahun Shibru Deyyasso

A Thesis Submitted to

The department of Electrical Power and Control Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters of in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

December, 2022

Adama, Ethiopia

Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)

Tilahun Shibru Deyyasso

Advisor: Prof. Seung Num Kim
Co-Advisor: Tamiru Getahun

A Thesis Submitted to
The department of Electrical Power and Control Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Masters of in Electrical Power and Control Engineering (Control Engineering)

Office of Graduate Studies
Adama Science and Technology University

December, 2022
Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled **"Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)"** is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

_____	_____	_____
Name of the student	Signature	Date

RECOMMENDATION

We, the advisors of this thesis, hereby certify that we have read the revised version of the thesis entitled **"Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)"** prepared under our guidance by **Tilahun Shibru Deyyasso** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Electrical Power and Control Engineering.

Therefore, we recommend the submission of revised version of the thesis to the department following the applicable procedures.

_____	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-advisor	Signature	Date

APPROVAL PAGE

We, the advisors of the thesis entitled "**Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)**" and developed by **Tilahun Shibru Deyyasso**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Advisor	Signature	Date
Co-advisor	Signature	Date

We, the undersigned, members of the board of examiners of the thesis by **Tilahun Shibru Deyyasso** have read and evaluated the thesis entitled "**Design and Implementation of Intelligent and Autonomous AI Operated Humanoid Robot (DINKINESH)**" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Electrical Power and Control Engineering.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
Dr Beteley Teka		26-12-2022
External Examiner	Signature	Date

Final, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

First and foremost, I am extremely grateful to my almighty of God, for his undiscoverable and unimaginable helps of being to be existed in the time of trouble for his hands always help my life without my understanding. Secondly, professor Seung Num Kim has been a model professor, mentor and thesis supervisor, offering advice and inspiration with perfect combination of insight and humor. I'm proud of, and thankful for, my time working with professor S.n. Kim.

I would like to thank my co-advisor Tamiru Getahun for his great support and intuitions foremost to the writing of this paper. My honest thanks also go to the HoD of ASTU EPCE mesifine megra and all of the EPCE staff for their patience and understanding during the two years of effort that went into the production of this paper.

A distinct thanks also goes to Dr. Endalew, from his experience many of the skill I gained to use in this thesis research paper I have been made.

This research work, would not have been possible without the constant financial support, guidance and the assistance of major advisors S.n. Kim. and my lifetime friend Henok Temesse, Kassahun Legesse, Mintesinot Legesse, their levels of patience, knowledge, and ingenuity is something I was always keep aspiring to.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES AND ILLUSTRATIONS

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
ASIMO	Advanced Step in Innovative Mobility
CoG	Center of Gravity
CoM	Center of Mass
CoP	Center of pressure
CPU	Central Processing Unit
DARPA	Defense and Research Project Agency
DDPG	Deep Deterministic Policy Gradient
DOF	Degrees of Freedom
EOD	Explosive Ordnance Disposal
EZMP	Extended Zero Moment point
FSR	Force Sensitive Resistors
FZMP	Fictitious Zero Moment point
GCoM	Ground Projection of the Center of mass
GPS	Global Positioning System
HAL	Hardware Abstract Layer
HIS	Hue, Saturation, and Intensity
IBM	International Business Machine
IMU	Inertial Measurement Unit
iPRAW	International Panel on the Regulation of Autonomous Weapons
LDUUV	*Large Displacement Unmanned Undersea Vehicle*
LIPM	Linear Inverted Pendulum
MAV	Micro Air Vehicles
MIT	Massachusetts Institute of Technology
Mph	Metter per-hour
NASA	National Association of Space Agency
RGB	Red, Green, Black
RRT	A rapidly exploring random tree
RUR.	Rossum's Universal Robots
SVM	Support vector machine
SVPP	Support Vector Machine-based path planner

UAVs	Unmanned Flying Vehicles
UCAV	Unmanned Combat Air Vehicles
UMSs	unmanned maritime systems
UUSs	unmanned underwater vehicles
VCP	Virtual Contact Surface
VFH	Vector Field Histogram
ZMP	Zero-Moment Point

ABSTRACT

The premature possibilities of the impression of machine intelligence systematized do not contain the partition of the emerging field of Artificial Intelligence. The organizers of AI proposed the concept of embedded intelligence as being adjoined between perception, reasoning and actuation. Until now the fields of AI and Robotics drifted apart. Professionals of AI focused on problems and algorithms preoccupied from the real world. Roboticists, generally with a background in mechanical and electrical engineering, focused sensory-motor functions. That departure was gradually being joined with the advancement of both fields and with the rising interest in the autonomous systems. The design and application of intelligent and autonomous AI operated humanoid robot (Dinkinesh) with the design and implementation of bipedal locomotion motion and balancing strategy for humanoid robot, path planning, design and integrated artificial intelligence. In the locomotion and balancing strategies, the dynamic modeling in the inverse kinematics problem with robot frame singularity, this can be improved by closed form inverse kinematics, with two stability measurements like, ZMP and COM based on MPC controller. In the conditions when the biped-robot operate in some useful operation, the path trajectory in accordance with the environments from the initial point to the goal point for those task the model predictive control with simultaneous gait pattern generation and path planning was implemented and obtained successfully. In the case of unstructured environment in the conditions of what a controller not understand what to do in known and unknown environments through the method like deep reinforcement learning, with the agent deep deterministic policy gradient (DDPG) which was just general artificial intelligence. The report introduces a method of implementing of intelligent and AI-operated humanoid robots Dinkinesh with a several capabilities of design parameters, tools, and techniques was used to integrate the robotic software architectures with its hardware system that allow a humanoid robot Dinkinesh to become more autonomous, intelligent and AI-powered. To cope with this issue the global perspective on main characteristics, design choices and constraints was briefly explained and the methods are utilized and the required output was discussed based on the hardware design requirements. The works had used their own the hardware design method from the scratch and all the material used to design the Dinkinesh biped humanoid robot was tested with postural stability and inverse kinematics problem was tested accordingly and gives good results.

Keywords; Autonomous, Intelligent, Artificial Intelligence, Humanoid Robots

CHAPTER ONE

INTRODUCTION

1.1 Background and thesis Overview

No matter how one believed in God's creations or just the theory of evolutions, it was clear that everyone that agreed on that the gifts of nature were attractive and just perfect. We are surrounded by highly developed natures that are fully adapted to their environment and able to react by any kind of environmental influences (Horakova, 2006). What we can to take granted by the nature is extremely difficult to realize in some technical systems. For instance, the complication of everyday tasks like "going straight on put one foot in front of the other ... just walk!" is anything but simple! Accordingly, the interaction between the "walking tool" leg and the body is highly complex: the body must be kept in balance at any time; environmental influences such as wind or the nature of the walking surface must be taken into consideration; the goal is to be focused, potential obstacles must be identified and alternative paths must be assessed.

For the first time, the term "robot" (Czech for "worker") the concepts were invented in 1920 by the Czech science fiction writer Karel Čapek. Karel Capek introduces the satires of the technological evolution in robotics as the first robots was designed like humans and it was developed for industrial factory. However still after one hundred years later the development of human like robots is just rare, although the number of research groups worldwide has increased exponentially during the recent years (Horakova, 2006). But, according to a study of the international federation of robotics only about 100 are in use, most of them were still in research laboratory (World Robotics, 2008). So, after all development in the research and investigation today's researchers or some scholars give a meaning to robotics and artificial intelligence (AI).

In this essence the Czech Karel Capek (1890-1938) wrote the play "R.U.R. – Rossum's Universal Robots" (World Robotics, 2008), that was created the foundation for the modern science fiction literatures. He created the term "robot", which was derivative from the Czech words "robota", denotation necessary labor, and "Robotnik", a term for a peasant owing such work. Moreover, R.U.R. ties the term robot to the vision of an intelligent artificial being with a mind of its own. This visualization was necessitated by the fear of men being suppressed by machines with superior physical and intellectual controls over mankind. In a reflective

examination it was stimulated that most science fiction authors evaluated the growth to be faster than it turned out to be. Previously in Japan, the first humanoid robot whose name is WABOT-1 was developed in 1973 at waseda university (Lim et al, 2002). After then, many humanoid robots have been developed continuously. For example, there are ASIMO of Honda (Hirai et al, 1998) (Sakagami et al, 2002), SDR series of Sony, H7 of Tokyo university (Kagami et al, 2002), HRP of AIST and so on.

What is Humanoid Robot?

A humanoid robot can be defined as "

… A robot with its overall appearance based on that of the human body. In general, humanoid robots have a torso with a head, two arms, and two legs (although some forms of humanoid robots may model only part of the body) … A humanoid robot is an autonomous robot because it can adapt to changes in its environment or itself and continue to reach its goal" (Science Daily: Humanoid Robot). In other side when we consider "modern" robots, in which numerically functioned and docile through the invention of George devol which was called the Unimate. The purpose of creating Unimate was for it to be applied within the automotive industry as a tool that would aid eradicate some of the dangerous and energetic work current employees were doing. To lift hot parts of metal from a die-casting machine and load them. Humanoid robots are now actuality applied in a widespread series of manufacturing. In terms of a product that is accessible to clients, Sony developed a robot named Qrio which dances, runs, recognizes faces, maintains its balance, and can get up if knocked over.

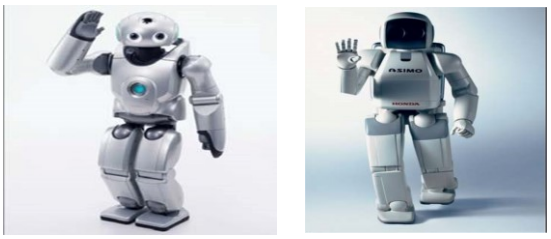


Figure 1.1: Commercial Robot (Qrio) vs Commercial Robot (ASIMO) (Alcaraz et al, 2013)

In the work force humanoid robots are currently a couple popular uses that will eventually be expanded upon. So, first of all, the humanoid robot must walk autonomously to live with a human. This can be possible by the ways of the humanoid robots can be programmed to perform almost any of the human actions without the need for any changes. The potential

behind using such robots is that it could substitute the human labor in dangerous and risky tasks such as firefighting or mineral mining. Therefore, the biped humanoid is essential to have a consistent, fast locomotion and the robust adaptive balancing controllers to accomplish the goal. In this a new inspiration of analysis of the biped stability locomotion by (Vukobratovic, 1990) that was developed by M. Vukobratovic and his all the team was organized to the problems generated by the functional rehabilitations around 1972 the first terms ZMP was exhibited.

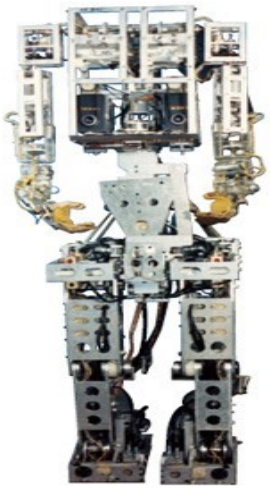


Figure 1.2: WABOT-1 robot (Lim et al, 2002).

That was the first trials in the formalization that was required for the dynamical stability of biped humanoid robots. So, the first idea was used in the dynamical wrench in order to improve the classical biped balancing stability criteria. For this concern the COM should be projected inside the convex hull of the foot contact points. Such that the ZMP concept was used in the coming generation of biped walking humanoid robots to made more robust and able to walk in the regular walking terrain, The ZMP concept was used in the next generation of walking robots, more robust and able to walk in irregular terrains. To design such an advanced biped walking humanoid robotics system, it should consist of a certain human action in the certain field that should perform a human action with the aid of path planning, obstacle detections, image processing, obstacle avoidance and etc.

Obstacle recognition and evasion are also a grand challenge in robotics control with navigation. One of the ways of detection of obstacles was the use of image recognitions. When we say using the image recognitions, the world map can be identified in the area with obstacles and without it was used as the most of research study nevertheless accept that the humanoid robots could capable served as a service robots that could leads to biped humanoid robots as an emerging field in the coming decades (World Robotics, 2008) (InteRegele et al,

2004). Despite widespread investigate in the field of biomechanics and motion sciences that could bring the answers for the question that how the humans solve the highly multifaceted problem of 3D BL with nonparallel robustness still not understated in present days.

Autonomous system: - an autonomous robotic system or function is just a closed loop system which integrated the capability of sensing, thinking and acting that could give machine to become with super automatic and super intelligent through the sensors used to receive the information from the environments like sense then it would process through like thinking finally it just act to the given environment dynamics accordingly. So, the autonomy is just the ability of the system to act without direct human connections although to act with various levels an many applicable areas.

Intelligent system: - intelligent system also can be described with the control system which is described by the computation efficient procedures of translating a highly complex system with the incomplete and in unstructured representation and under incomplete specification of how to do something in the certain environments towards to a certain goal (Sun, 2007). Generally, the intelligent control system mainly has their own the following characteristics that should contained of learning functions, adaptive functions and organizing functions by using the theories from the mathematical and seeks inspirations and ideas from the biological systems.

Artificial intelligent system: - the term AI that was first coined in the Dartmouth conference at 1956 that was organized Marvin Minsky, John McCarthy and two senior scientist, Claude Shannon and Nathan Rochester of IBM. At that conference the expressions of AI were for the first time it was coined in the title of the field. Such the well-known definitions were described in the table below.

Table 1.1: Some definitions of artificial intelligence, organized into four categories

Thinking Humanly Thinking Rationally	**Thinking Rationally**
➢ *"The exciting new effort to make computers think . . . machines with minds, in the full and literal sense."* (Haugeland, 1985) ➢ *"[The automation of] activities that we associate with human thinking, activities*	➢ *"The study of mental faculties through the use of computational models."* (Charniak and McDermott, 1985) ➢ *"The study of the computations that make it possible to perceive, reason, and act."* (Winston, 1992)

such as decision-making, problem solving, learning . . .” (Bellman, 1978)	
Acting Humanly ➢ *“The art of creating machines that performs functions that require intelligence when performed by people.”* (Kurzweil,1990) ➢ *“The study of how to make computers do things at which, at the moment, people are bette*r.” (Rich and Knight, 1991)	**Acting Rationally** ➢ *“Computational Intelligence is the study of the design of intelligent agents*.” (Poole *et al.*, 1998) ➢ *“AI . . . is concerned with intelligent behavior in artifacts.”* (Nilsson, 1998)

A practical definition to say a given system optimized in the certain level of human intelligence those introduced by Russell and Norvig; the AI is the study of human intelligence and actions replicated artificially, such that the resultant bears to its design a reasonable level of the rationality (Russell & Norvig, 2009).

1.2 Statement of the problem

Humans have continuously pursued to imitate the appearance, mobility, functionality, intelligent operation and thinking process of biological creatures. In the recent progress sought a significant advance have been made in making sophisticated biologically inspired system, using these advances, scientists and engineers are increasingly reverse engineering many animals’ capabilities.

Progress in artificial intelligence, artificial visions and many other biomimetic related fields are leading to many benefits to mankind. Such a complete design of humanoid robots contained of a several modules inside in it; like designing the mechanical models, integrating the mechanical models into the control system at the end the robots to perform walking and balancing, path planning and obstacle detections, motion planning, human interaction, face recognition, speech recognitions and etc.

In the biped locomotion and balancing strategies, the numerous gaps were still in progress, amongst of that’s the dynamic modeling in the inverse kinematics problem in most of paper based on the analytical Jacobian, which leads this to singularity in the robot frames, this can be improved by closed form inverse kinematics, in other context no matter what type of controller we used to control the locomotion and balancing strategies in the biped walking

robot the fundamental concern in the stability of the gait relied on at the minimum with two stability measurements like, ZMP and COM of the robot and most of the stability constraint was limited only in one variable this was just not enough in the solving all the stability constraints, this can be achieved through the utilization of multiple stability measurements like COM and ZMP. To model the biped locomotion and balancing strategies that was based on the model predictive control within the compromise of low-cost RC servo motor for the position control instead of torque control or indirect control. In the conditions when the robot operate in some useful operation, they must be planning the path in accordance with the environments from the initial point to the target goals point for that task the model predictive control with simultaneous gait pattern generation and path planning. In the case of unstructured environment in the conditions of what a controller used in the biped humanoid even not understand what to do in known and unknown environments.

To answer these question the integration of AI to robotics plays the other great roles by using method like deep reinforcement learning, this because of the capability deep reinforcement learning, can be able to used instead of the control theory in the continuous state actions pairs, this was allowed through the deep deterministic policy gradient(DDPG) that's was just general artificial intelligence, the basic requirements in the integration of AI method in the biped walking humanoid robot were basically to achieve in the replacements of the conventional control theory with a simple and efficient AI method so as to reduce the complexity in the mathematical derivation, more robust and to made more intelligent. This reinforcement problem can be represented with discrete or continuous state space action within a model based or model free dynamics. The primary aim of the study was laid on the fundamental research questions of the thesis; this was a relevant design and implementation in locomotion and balancing strategy, path planning, integration of robotics and artificial intelligence together. In this study the best of all platforms to integrate the robotics and artificial intelligence together by applying the relevant control theory, mechanics of humanoid robots, and by applying AI method in to Dinkinesh humanoid robot was assessed.

1.3 Motivation of the project

The study of humanoid robots are the most complex and multidisciplinary research study agendas with an integrated procedure of mechanics, control, computer science, mechatronics, all the field work together, nevertheless the advancement become more capable than the former generations that because of the availability of newer technologies of newer technologies at realistic and reasonable costs. The technology should powered that to

included more capable batteries, the powerful computers with the lower power consumptions and more reliable actuators. These involvements are improving the humanoid robots research and challenges with the able subject filed matter. However, the specific environments would require lots of improvements to be suitable for the wheeled robots (Marc, 2006). surpass on surfaces like roads, present poor mobility in uneven or spongy terrains, thus why present poor mobility in uneven or spongy terrains, thus they can only access half of the earth's landmass. Huang et al. stated that that biped robot possesses higher mobility than wheeled robots, especially when moving in rough terrains (Huang et al, 1999). Humanoid robots are designed with a structure of the similar to the humans and are ready to be programmed to perform most of the human's tasks. Since most of the work-suitable robots are expensive, it is expected that industries would gradually switch its workforce to robots.

1.4 General and Specific Objectives

1.4.1 General Objectives

- Design and implementation of intelligent and autonomous AI operated humanoid robot (Dinkinesh)

1.4.2 Specific objective

- To design and implement a bipedal locomotion and balancing strategy for humanoid robot.
- To design and implement a path planning for biped humanoid robot.
- To design and integrate biped humanoid robot and artificial intelligence.

1.5 Significance of the study

Currently, the robots performed a number of different jobs in numerous fields and the numbers of tasks delegated to robots is rising progressively. So that the best ways to splits robots into several types is a partition by their applications or just by the significance of their tasks.

1.5.1 Robots in medical industry

The existing technology are being combined in new ways to streamline the efficiency in the health care operations as the results a wide range of robots was being developed to serve in a variety of roles in the medical industry from this one the Telepresence the one that the physicians used with helps of robots helps some of medical operations through the remote locations in order to help the patients from the remote operations and the other one is the ***surgical assistants***:- these remote controlled robots to assisted surgeons with the performing

operations, typically minimally with invasive procedures. The rehabilitation in the other concerns this plays a crucial role in the recovery of people with the disability, including improved mobility, strength, coordination and quality of life. Sanitation: with the increase in antibiotic-resistant bacteria and outbreaks of deadly infections like Ebola, COVID-19 and more healthcare facilities are using robots to clean and disinfect surfaces.

1.5.2 Robots in military industry

The most of the robots that are mainly employed within an integrated system that include of the video screens, sensors, grippers and cameras and as it is the military robots also should capable to have different shapes and sizes so that the size according to their purpose and they may be either an autonomous or remote-controlled devices. For this regards the most important dilemmas of fighting robot can be held in the fighting robot categories ‘or other principles also they divided within their various political, economic interests and the level of knowledge.

Ground robots; this robot has the general capability of the robots can save the lives which has secured future path for the ground robots alongside the warfighter. So, the ground robot can be engaged in different missions including explosive ordnance disposal (EOD), combat engineering reconnaissance and many other of its forms. The boston dynamics designed the LS3 that was a robot mule to help soldiers carry heavy loads, see *fig 1.3* in the middle the LS3 in a rough terrain robot designed to go anywhere.

Figure 1.3: The boston dynamics mules; carrying heavy loads following soldiers; moving through the complex terrains (Boston Dynamics, 2014).

As shown in the figures in *fig 1.4,* is also the fastest legged robots in the world, with its own the maximum distance to travel 29kmph that was the first world recorded legged robots in the world, it was the new land speed record for legged robots (Boston Dynamics, 2013). The cheetah robot has an articulated back and forth on each step, increasing its stride and running speed as much like as the animals does.

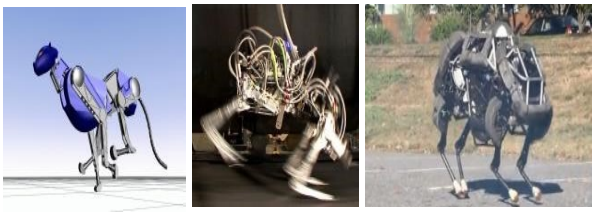


Figure 1.4: Boston Dynamics robots: The Cheetah concept; Cheetah on a high-speed treadmill; Cheetah becoming Wild Cat running untethered (Boston Dynamics, 2013)

Aerial robotics; The US army, the air forces and the navy had developed a variety of robotic aircraft known as unmanned flying vehicles (UAVs). As like the ground vehicles, those robots had the dual applications, they can be used for the reconnaissance without endangering human pilots, and they can carry missiles and other weapons. **Maritime robots** also the other one that applied in the sea-based robots unmanned maritime systems, or UMSc, can be either free-swimming or tethered to a surfaces vessel, a submarine, or larger ranges of robots that tethers simplify providing power, control, and data transmission, but limit maneuverability and range.

Figure 1.5: a) Swarm of autonomous boats; b) Harvard University multiple robots operating without central intelligence; c) Sci-fi image of future robotic armies (Hsu, 2014)

In the same manner Harvard university scientist have devised a swarm that of 1024 tiny robots that can work together without any guiding central intelligence.

1.5.3 Robots in entertainment industry

One of the most applicable areas of biped humanoid also in the entertainment industry in which a humanoid robot can take the form of interactive communications marketing tools at the trade bazar such that the promotional robots move about the trade shows a floor that providing tongue in cheek interactions with the bazar to bring customers to a particular company. These promotional robots are hired by the company to entertain the information in the trade shows the attendants introduced the product and service that was available in the trade shows. For example, a well-known Japanese, firms sell as of now 15,000 in the entertainment robot industry at the rather and higher price which nevertheless only reflects

the integrated expensive hardware, with a very effective marketing policy creating tenfold over the demands.

1.5.4 Robots in space exploration industry

Robots are most widely used in space search. It can easily work in harmful space where human being can't perform, in this case many robotic developer, manufacturer and researchers were dedicated in the robotic spacecraft design to make scientific research measurements is often called a space probe. Many space missions are more suited to telerobotic rather than in the crewed operations, this is because off lower costs and lower risk factors Robonaut is one of space robots in DARPA-NASA project that designed to create a humanoid robot which can be functioned as the equivalent to humans during extra vehicular activity (space walks) and exploration.

1.6 Scope of the study

In this thesis scope research was focused on the design and implementation of autonomous and intelligent AI powered humanoid robots(Dinkinesh) in the aspects of, the design and implementation of the locomotion and balancing strategies of the humanoid robots which mean that the walking and balancing control; this was totally all about to design a biped locomotion approach, path planning system of the robots robot this allows a robots to reach from the initial point to goal point and the design to integrate robotics and artificial intelligence together this one allows to Dinkinesh to behave like human to set in the certain degree of autonomy.

1.7 Thesis Organization

The thesis was organized with six chapters, the chapter one; introduction which describe the introductory part of the paper with description of background and the thesis overview, the problem statements of the thesis, which is describing the fundamental problem of the study gaps, the motive of the thesis, the objectives of the thesis, the significance of the study, the scope of the study and finally the paper organization.

Chapter two, Literature review; which contained of literature review, the chapter described some useful theoretical description of the study according to the specific objectives of the study this includes of locomotion and balancing strategies of the humanoid robot, the path planning control and finally the review of the integrated robotics and artificial intelligence.

Chapter three, Methodology and mathematical modelling, in this chapter the proposed method and the modelling of the biped locomotion and balancing strategies of Linear

inverted pendulum, model predictive control, path planning and obstacle detection system and finally the mathematical modelling of the deep deterministic policy gradient were described according to the thesis specific objectives and also the hardware system design and integration and the material which used in the thesis were described.

Chapter four, simulation results and discussion, in this chapter the simulation of the software system of the study held down, this simulation results of the biped locomotion and balancing strategies, designing of the path planning system and finally the designing of the artificial intelligence method were discussed accordingly.

Chapter five, Hardware design discussion and results, in this chapter the hardware design of the Dinkinesh biped humanoid robot were described in such that; it contained of the mechanical design and assembly process and the electronics prototyping and PCB design process were described with the objectives of the thesis which modeled dynamics of the biped locomotion and the balancing strategies, path planning and obstacle detection and finally the integration of the robotics aspects with the artificial intelligence were tested.

Chapter six, conclusion and recommendation, this chapter describe about the final conclusion and recommendation of the study according to the specific objectives and general objectives of the study.

CHAPTER TWO

LITERATURE REVIEW

2.1 Locomotion and Balancing Strategy of humanoid robot

2.1.1 Locomotion and Stability Measurements

The humanoid robot is notable by their capability by the moving using its two legs and this locomotion's characteristics made the biped robots could capable to work in any human-suitable environments without the need to changes any situations, but the wheel driven humanoid robots are more widespread than biped humanoid because of its more stable and controlling its motion is much easier; however, it cannot move over objects or climb a stair as like biped humanoids.

However, the complex and non-linear structures of the biped humanoid made it to be susceptible to lose the balance while in motion. Such that the balancing during the locomotion remains in active area of research. The stability of the biped humanoid gait can be evaluated by measuring the relative distance between specific points on the ground with the several stability measurements variables. To implement the biped walker, it was essential to determine whether the robot is in danger of tilting and other fall parameters, so that the mathematical criteria for this property are reviewed in the sections below within how to achieve the stability of the gait anytime.

2.1.1.1 Center of Mass

The practical definitions COM of the articulated mass body is the unique point where the slanted virtual location of the distributed mass sums to zero (Wikipedia, 2021). The center of mas also the term used for the motion visualization of the multi-link interconnected mechanics of humanoid. Within the regards to the masses of the rigid body dynamics because of it has to be this COM as the reference point for mechanical calculations, so that the COM of multi-linked body in the 3D can be calculated in the equation below 2.1 (Wikipedia, 2021).

$$x=\sum_{i}^{N}\frac{M_i x_i}{M}, y=\sum_{i}^{N}\frac{M_i y_i}{M}, z=\sum_{i}^{N}\frac{M_i z_i}{M} \tag{2.1}$$

Were as the variables of x, y, and z being the 3D cartesian coordinates of the COM X_i, Y_i, Z_i, are the cartesian coordinates of the CoM of the *ith* link. m_i represents the mass of the ith

link *M* is the total mass of the body and *N* are the number of the body links (Wikipedia, 2021).

2.1.1.2 The Ground Projection of the Center of Mass

Stationary robot can only experience the gravitational forces which are applied on the all parts of the robots, such that this force can be replaced by the virtual forces acting at the (CoM) of the robot, whereas $m_{i,}$ denotes the mass of ith links of the robots and p_i the position of its center of mass. The location of the COM is decisive for the equilibrium of the robot. Its orthogonal projection from the ground is commonly referred to as the ground projections of the center of mass (GCoM) (Kajita et al, 2003).

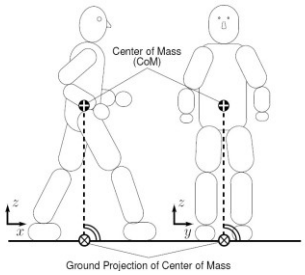


Figure 2.1: CoM and its ground projection (Shafii, 2015)

So generally, the GCoM is used to evaluate the static stability of the gait as it will explain latter such that the horizontal GCoM for the multi-linked system is given by the equations 2.2 and 2.3.

$$x_{GCoM} = \frac{\sum_{i=1}^{n} m_i X_{ci}}{\sum_{i=1}^{n} m_i} \tag{2.2}$$

$$y_{GCoM} = \frac{\sum_{i=1}^{n} m_i Y_{ci}}{\sum_{i=1}^{n} m_i} \tag{2.3}$$

- *mi* represents the mass of the ith link, X_{ci} and Y_{ci} are the position of the center of mass for the *ith* link.

2.1.1.3 Zero moment point

One of the most important notable criteria of the biped humanoid robot is also the zero-moment point, this criterion is also can be defined as it's the point from the ground relative to the COM that will generate an inertia that become zero due to the gravity(Goswami A. , 1999) (Vukobratovic, 1990). So generally, the ZMP is the point in the biped humanoid robot

is that commonly can be affected by the links of the robot's position and inertia. Which repeatedly changes during the motion state. The zmp of a multi-link body can be calculated by the equations of (2.4) and (2.5) (Shafii, 2015), where P_x and P_y also the position of the ZMP in the Cartesian domain.

$$P_x = \frac{\sum_{i=1}^{n} m_i\left(\ddot{z}_i + g\right)x_i - \sum_{i=1}^{n} m_i \ddot{x}_i z_i - \sum_{i=1}^{n} I_{iy}\ddot{\Omega}_{iy}}{\sum_{i=1}^{n} m_i\left(\ddot{z}_i + g\right)m_i} \tag{2.4}$$

$$P_y = \frac{\sum_{i=1}^{n} m_i\left(\ddot{z}_i + g\right)y_i - \sum_{i=1}^{n} m_i \ddot{y}_i z_i - \sum_{i=1}^{n} I_{iy}\ddot{\Omega}_{iy}}{\sum_{i=1}^{n} m_i\left(\ddot{z}_i + g\right)m_i} \tag{2.5}$$

In the above equation, x_i, y_i and z_i also the CoM vertexes of the ith link in mi of mass associated from the ith links within the gravitational constant g within moment inertia and angular I_{ix} and $\ddot{\Omega}_{iy}$ about the X-axis, respectively.

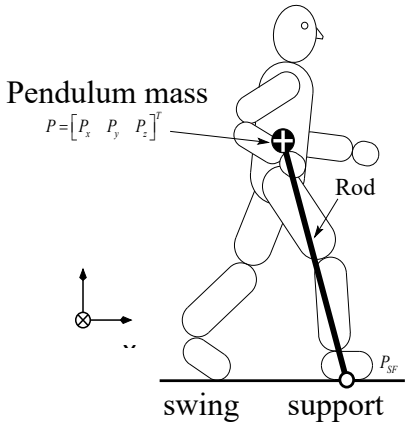


Figure 2.2: The zero-moment point reaction to COM (Novacheck, 1998)

Extended zero moment point: the EZMP was proposed by (S. G, et al, 2015) as an extension to the concept of zero moment point. The ZMP of the robot cannot be defined if the robots' feet in contact with surfaces on different planes.

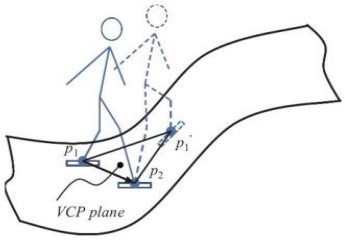


Figure 2.3: The virtual contact surface (S. G, et al, 2015)

2.1.1.4 Center of Pressure

The center of pressure (CoP) can be defined as appoint on the ground that the ground reactions forces act (Shafi, 2015). The CoP is just the point where the resultant moment generated by inertial to the ground and the net moment in the horizontal generated by inertia and gravity forces is tangential to the ground and the net moment in the horizontal direction were resulted to zero (Goswami A. , 1999). So that during the gait, there are also two types of the force it could associated with that was also contact and non-contact forces, so that CoP is related to the first type, while the ZMP is linked with the second (Sardain & Bessonnet, 2004). So, what's the main differences between the CoP and the ZMP is that the CoP and the ZMP is that the CoP never leaves the area covered by the robot feet, while the ZMP can temporary leave that area in the CoP of multi-linked body could be calculated using the equation of the (2.6) (Shafii, 2015)

$$OP = \frac{\sum_{i=1}^{N} q_i F_{ni}}{\sum_{i=1}^{N} F_{ni}} \tag{2.6}$$

2.1.1.5 Support Polygon

Many of the stability method used for the foot support area in the measurement approaches that was called support polygon (Shafii, 2015). So actually, the support polygon is also the area covered by single foot or the convex hull of two feet area in single and double support phase respectively.

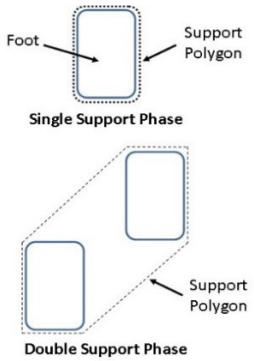


Figure 2.4: Supporting Polygon (Brehm, 2017)

2.1.1.6 Static and Dynamic Stability

In order to biped gait is said to be statically stable and a posture is said to be balanced if the gravity line from its center of mass falls within the convex hull of the foot support area (Goswami A. , 1999).

So, the common stability criterion that the robot remains in balance will not achievable during falling while at the locomotion, so that one drawback in using this stability measurement assessment that is it cannot distinguish between the marginally stable and in unstable state thar was because of in both cases the ZMP is located at the polygon boundary (Shafii, 2015) .

2.1.1.7 Summary

Improving the biped locomotion stability remains in the one of the active research areas in the humanoid robotics world, that is because of the biped non-linear structure, balancing the locomotion's is a non-trivial task. It is essential to understand the concepts behind the stability. In this chapter, I have introduced and illustrated various concepts related to locomotion and the balance measurements stability, such as the center of pressure, the ZMP, the support polygon and the center of mass.

2.1.2 Biped Locomotion Approaches

2.1.2.1 Overview

The biped locomotion also signifies a humanoid robots moving mechanisms using its legs. And this locomotion is one features that is special with humanoid robots moving mechanism using its legs and this locomotion is one features that is special with the humanoid robots, the questions was there are various robotic locomotion approaches that developed within the different aspects. In this section, several factors that can be used to classify the walking approaches are discussed and the control algorithms that is going to be used are discussed.

Passive Vs Active Walking

Passive or cyclic walking can be defined as a robot a natural walking using only gravitational forces without using any power or joint actuation. Passive walking can be summarized walking using only gravitation forces without using any power or joint actuation. Passive walking can be also summarized as unaided walking down a slight inclination plane (Arbulu, 2009).

Static locomotion Vs Dynamic Locomotion

Static locomotion is achieved based on the CoM position only without considering the CoM acceleration, as we can stated a static walker is assumed to remain in balance if the GCoM is inside the support polygon all the time. So, the CoM trajectory for this approach is almost the same as the ZMP (Shafii, 2015). Graf and Rofer also stated that the static closed loop walking approach for soccer was use the CoM trajectory (C. Graf and T. Rofer, 2010). The

static gait is much slower than a dynamic gait because it restricts the GCoM from the leaving support polygon, which made it less popular than the dynamic locomotion.

2.1.3 Biped Locomotion and Balancing Control Strategies

Biped locomotion and balancing control are the most important concepts in the design of intelligent and dynamic walking for the humanoid robots, in this regard to work with the dynamic locomotion and stability, the model of the humanoid robots depends on several design parameters which lead to one of the list of the final control elements, so in this concern of the design of bipedal locomotion possibly can be designed and modeled with a several physical system model like, Cart table, 2D/3D inverted pendulum and so on.

2.1.3.1 Cart Table module

The cart-table physical system model that simplifies the robot's dynamics and provides a mathematical relation between the CoM position and the ZMP, cart table module can relate the humanoid robot center of mass and the zero moment point of the foot. As shown in figure 2.5, the humanoid walking can be represented by cart-table model during the single support phase. The model assumes all the masses are concentrated in a cart and the robot support leg is massless.

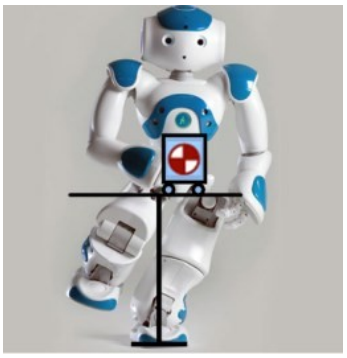


Figure 2.5: The Cart-table Model for the NAO (Alcaraz et al, 2013)

The principle of the model is that it balances an object by supporting its center of mass and the ZMP can move in the foot which is modeled by the table contact with the floor. During locomotion the CoM moves along both planes (Alcaraz et al, 2013). If the CoM position along a plane exceeded the supported area, then a horizontal moment will be exerted on the axis and the table would topple. *Figure 2.5* provides a schematic view of the model.

2.1.3.2 Inverted Pendulum Mode

The linear inverted pendulum (LIPM) is another physical model that can be used to generate humanoid motion (Katija et al, 2010). It is an extension to the cart-table and has the same

equations. As in the cart-table, the LIPM simplifies the robot as a single mass point on its CoM (Katija et al, 2010). The LIPM assumes that the ZMP were at the origin O, which corresponds to the robot's ankle, and the ankle's torque just gone to Zero (Arbulu, 2009). In contrast with LIPM, the cart-table model allows the ZMP to move in the foot that is modeled by the table contacts with the ground, and the torque of ankle is not required to be zero (Choi et al, 2004). The LIPM also has the drawback of not considering any angular momentum associated with the motion.

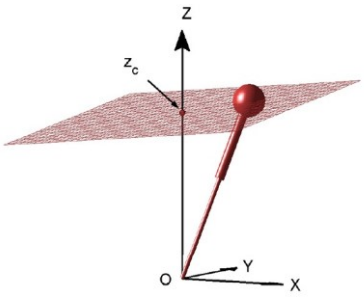


Figure 2.6: Linear Inverted Pendulum (Kajita et al, 2003)

2.1.3.3 Gait Analysis

In this section, I present and define several leading terms that are vital to understanding the development of a walking engine.

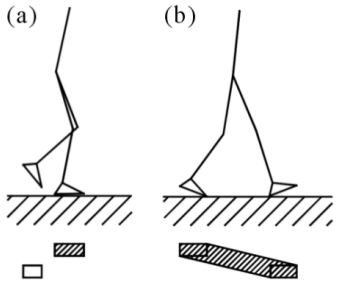


Figure 2.7: Support Polygon during SSP (a) and DSP (b) (Shafii, 2015)

Double Support Phase (DSP): - describe as an instant where both of the robot feet are in contact with a surface on the floor. When a biped is in double support phase, it is supported by both of its feet, i.e., the two feet are on the floor surface or on an object that is placed on the floor.

Single Support Phase (SSP): - Define a situation when one the robot feet are in constant with a surface on the floor. During SSP the biped uses only one foot to support its weight. DSP put the robot in a more stable state as the support polygon area is enlarged when the two feet are in contact with the floor. Figure 4.1 shows both the DSP and SSP and its relation to the support polygon area.

2.1.3.3.1 Swing and Stance Leg

The Swing leg is the leg that is performing a step by moving or swinging forward through the air. In contrast, the stance leg denotes the leg that is fully supported by the floor and supports all the robot's weights.

2.1.3.3.2 Single Direction and Omnidirectional Walking Engine

Single direction walking engine

A single direction walking engine allows the robot to walk in straight lines. This approach is not very practical for humanoid robots in dynamic environments, because the robots need to change its walking direction continually as the surrounding environment changes.

Moreover, Humans continually changes their walking speed and direction in everyday tasks and since it is desired to have humanoids robots working at the human's level skills, improving the omnidirectional walking engine remains an active research area. As it was mentioned above, Erbatur et al. have used fourier series to represent the ZMP and solved the cart-table equations (Erbatur & Kurt, 2009).

His approach resulted in a single direction gait without the ability to perform a curved or diagonal locomotion; because they failed to connect the CoM trajectories smoothly while changing walking direction. The approach was improved later to generate curved walking. Ferreira et al. improved Erbatur method to generate diagonal walking, in which they decomposed the walking trajectory into time segments and added the double support phase to formulate the ZMP trajectory.

2.1.4 Biped walking Control Techniques

A bunch of controllers was applied in the existed development of humanoid robot in the case of walking control algorithm in this section, I had reviewed the existed literature that just designed for the humanoid robot based on the several control algorithms in the specified control objectives of the research domain in the given section below.

2.1.4.1 Model Predictive Control

To control the model of the biped humanoid robot in walking control and balancing strategies, many researchers around the world applied and had used the model predictive control. In the paper of MPC control in the underactuated biped walking robot in the presence of actuators toque saturation (Mathew et al, 2017), they had proposed method of synthesizes elements of the human-inspired control (HIC) approach for generating provably-stable walking controllers, rapidly exponentially stabilizing control lyapunov functions (RES-

CLFs) and standard model predictive control (MPC). An existing method of proving stability of the orbit rely on exponential stability of the continuous time controller, and to the authors' knowledge, a proof of stability under nontrivial torque saturation was still there an open problem (Mathew et al, 2017). The standard approach which was also need to be resolved the constrained nonlinear optimization for a different gait with lower torque bounds; however, this comes with its own challenges and it has to be done offline. Instead, the current MPC formulation intends to be a step towards handling (reasonable) torque bounds at the online control implementation level, and thus, relieve some of the burden of the offline optimization (Mathew et al, 2017).

In other paper of Marc et al; they had used the model predictive control in the legged robotics which was tried to be addressed with the three problems. Firstly, the reference trajectory was planned based on the stability analysis of the robot (ZMP/FRI) (Marc et al, 2018). Then it was a desirable to obtain a minimum jerk humanistic movement(Marc et al, 2018). They had presented work that proposed with a model predictive controller to regulate a highly nonlinear system, namely walking motion of a three-link humanoid robot with point feet by Marc et al, As the preliminary studies they revealed in their work the main problems of the kinds of MIMO systems which can be utilized in the biped walking humanoid are the strong coupling between the inputs and outputs. In order to have a stable walking biped one needs a master controller (Marc et al, 2018); however, in their paper by Marc et al, they hadn't mentioned the stability of the gait in their work as a main objective however the classical controls (computed torque with feedback linearization) along with the model predictive control were evaluated. The preliminary studies were necessary in order to compare those types of controllers which exist till date with the implemented one (Marc et al, 2018).

(Hong et al, 2020) also presented a constrained nonlinear model predictive control (NMPC) framework for legged locomotion. The framework they had assumed a legged robot as a floating base single rigid body with contact forces being applied to the body as external forces. Within the consideration of orientation dynamics evolving on the rotation manifold, analytic Jacobians which was necessary for constructing the gradient and the Gauss-Newton Hessian approximation of the objective function are derived (Hong et al, 2020). (Wieber, 2006) had also used a model predictive control, to control a biped walking robot scheme specifically which designed to deal with such a constrained dynamical system, with the potential ability to react efficiently to a wide range of situations by apart from the question of computation time which needs to be taken care of carefully (these schemes can be highly

computation intensive). A key idea for answering to this problem can be found in the ZMP preview control scheme, which can able to compensate a strong perturbation of the dynamics of the robot and weber proposed a linear model predictive control scheme which was an improvement of the original ZMP preview control scheme (Wieber, 2006). This question is of particular interest because of the limited horizon over which computations can be carried out: the problem was to find which optimal control problem can lead to stable long-term motions while being solved only over short horizons.

Table 2.1: Summarized literature review on MPC for the biped humanoid robot

References	Titles	Methodology	Contribution	Gap
(Mathew et al, 2017)	*Model pridictive contorl of underactuated bipedal robotic walking.* (Mathew et al, 2017)	Rapidly exponentially stabilizing control lyapunov functions (RES-CLFS) and standard model predictive control with linearization (Mathew et al, 2017).	The researchers also detected the formal stability of the formal stability of the hybrid periodic orbit as a result of the proposed MPC control approach which was a prime focus of existed research (Mathew et al, 2017).	They had used the dynamic model from the kinematics of Jacobian and most of the time the closed forms of IK problems was preferred in the design of humanoid robot, because of the kinematics singularity (Mathew et al, 2017).
(Marc et al, 2018).	*Model Predictive Control of Biped Walking on Humanoid Robot* (Mathew et al, 2017)	Feedback linearization along with computed torque (Mathew et al, 2017).	The evaluation of MPC in nonlinear MIMO in bipedal walking robot (Mathew et al, 2017).	They hadn't mentioned the stability of the gait in their simulation output (Mathew et al, 2017).
(Scianca et al, 2020)	MPC for humanoid gait generations stablity and feasibillty(Scianca et al, 2020).	They had used as prediction model was dynamically extended LIP with ZMP (Scianca et al, 2020).	They had produced a real time gait which includes of the footsteps with timing control (Scianca et al, 2020).	In their paper the stability constraint was linked only with the future ZMP velocities of an existed system model which just guaranteed the bounded CoM trajectory generation with respect to the ZMP trajectory (Scianca et al, 2020).

(Hong et al, 2020) al	Real-time constrained nonlinear model pridictive contol SO(3) Dynamic legged locomotion (Hong et al, 2020) al.	The analytic Jacobian with the gradient and the Gauss analytic Jacobians with gradient and the Gauss-Newton Hessian approximation of the objective function (Hong et al, 2020) al	The control mechanisms of the sophisticated dynamic locomotion of robot's under-actuated of the floated base that can only be controlled indirectly by the internal motion of the robot and external wrenches exerted on the robot (Hong et al, 2020) al.	The analytical Jacobian most of the time they are not suitable for the motion kinematics formulation in the complex humanoid robot thus because of the singularity and coordinate frame alignment (Hong et al, 2020).
(Wieber, 2006)	*Trajectory free linear model pridictive control for stabe walking in the presence of strong disturbances* (Wieber, 2006).	Linear MPC with ZMP preview control scheme Linear MPC with (Wieber, 2006).	Linear model predictive control scheme which was an improvement of the original ZMP preview control scheme (Wieber, 2006).	They had used indirect position control from the torque perturbation and it is not suitable for the RC servo motor and it needs the extra tachometer sensor reading (Wieber, 2006).

2.2 Path planning control of humanoid robot

The robot path navigation is the fundamental and mandatory scenario when we talk about a humanoid robot that was the one that had an interaction with an environment, so that this robot navigation includes of obstacle detection, path finding and obstacle detection, path finding and at the end the obstacle avoidance algorithms. According to the Kalah et al (Kalal et al, 2019) the implementation of the shortest distance path finding algorithm and the object detection avoidance using the image processing and artificial intelligence. Their works was mainly focused on generating shortest path using A* algorithm and object avoidance with image processing. The work also represented by inserting static map with the helps of the image labels.

The insertion of images object was tested with different shapes and size to be represented on the graph such that the shortest path will searched in the road networks with the A* algorithm in a road for the traffic congestions, in obstacle detection and path findings for the mobile robot and thus after through a survey in the field of obstacles and at the end system

implemented within two major modules that was the detections and avoidance scheme yet the feasible path into a single modules with the help of OpenCV library of python using the A star algorithm in the field of AI and image processing was performed in the study. In the similar manner in the other papers of the design of improved path finding algorithm that will find the near optimal path in a short time was proposed by Abiyev et al, (Abiyev et al, 2015).

Saeed and reforgiato (Saeed & Reforgiato, 2019) also presented a new off-line path planning method for the mobile robot to generate an optimal or the near optimal collision-free path between the initial points to the goal points in the given environment dynamics that contained of an obstacle. In the new method that called a boundary node method, the robot will be simulated by the nine-node quadrilateral elements, the centrode node represented by the robot location and it moves with the eight-boundary nodes in the working environments. So that the proposed method was capable of generating the initial collision-free path for the mobile robots safely and quickly.

Subsequently, an additional new method called path enhancement method was used to find shortest path by reducing the overall initial path length. The simulation results indicate that the method can successfully generate an optimal or near optimal collision-free path efficiently. In this paper a simple method of path planning was used with the help of zero moment point and model predictive control used in accordance with a different complicated path point of the environment dynamics.

2.3 Integrate Robotics and Artificial Intelligence

The paper by K. Rajan (Rajan, 2017) they also explored the integration of AI and Robotics in broader term and the key objective was; the critical aspects of AI techniques must consider in order to be applicable in to robotics system with the convenient way to group AI techniques according to the robotic challenge and the cognitive ability. The Hybrid reasoning**,** most of AI techniques address a single cognitive ability and rely on a single type of knowledge. This would need AI and Robotics to aid researchers in the pure (and social) sciences to obtain a finer level understanding of earthbound environmental processes.

In the paper by F. Ingrand and M. Ghallab, (Ingrand & Ghallab, 2017) the deliberation of autonomous robots surveyed based on, design of a systematic robotic deliberation, planning, acting, monitoring and goal reasoning, observing and acquiring deliberation models. So, the numerous deliberation systems were discussed along to their planning, acting, monitoring, observing and learning function. Most of the analyzed system that was addressed in the

three main function; planning, acting and monitoring. It was interesting to note that, in most of these systems, planning explicitly takes into account space and time.

In other side Agustin et al (Agostini et al, 2017). In their study, they showed that AI techniques can be efficiently integrated into the robotic platforms to learn and execute human-like tasks by exploiting natural teacher-learner interaction with the involvement of human interactions. The major features of the proposed method that allows correct post order to be found development of an efficient approach that permits the robot to determine causative conditions as well as the action that led to the observed changes.

The researcher also showed that the problem can be solved by using an easy-to-implement approach that evaluates the different combinations of conditions in parallel while considering the lack of experience, which was the fundamental for preventing biased estimates when learning based on the few trials. The fundamental contribution of the study was the development of a method that can help to make the robot to be the parts of our worlds, where they might help us the rapidly without requiring tedious programming or the complex interactions.

In the similar manners, the works Kunze and Beetz study(Kunze & Beetz, 2017). The envisioning the qualitative effects of the robot manipulations actions using simulation-based projection and also the main goals that was presented in their how the robot should capable reasons semantically about the objects and actions that rely on the richness of the continuous world. The embedded reasoning components deeply within robot control programs, allowing programmers to write more general control program in the concise ways. In thesis the walking biped humanoid robot with deep reinforcement learning algorithm were used in this regard, the deep deterministic policy gradient and the similar works of the other were examined in the section 2.3.1 below accordingly.

2.3.1 Deep reinforcement Learning

The deep reinforcement learning plays a vital role in the development of artificial intelligence method in the field of numerous realtime applications; this deep reinforcement learning may be expressed in the many forms within model based or model free, continuous action or the desecrate action and so on. In this study the development of DRL approach in the applications of the biped walking humanoid robot within the deployments of an agent called DDPG policy gradient agent functions so in the section below I had reviewed the related literature of other previous works.

Jun.M et al, also proposed that the model-based RL algorithm for the biped walking humanoid robot in which the robot learned appropriately for placing the swing leg, that to decide whatever action to perform based on a learned model of the Poincare map of the periodic walking pattern from the model they had used to maps from a state at the middle of a step and foot placement to a state middle of step within modified desire walking cycle frequency based on online measurements, the presented simulation results with an implemented approach of the actual biped robot (Jun. M et al, 2004). Their goal was to understand how humans learn the biped locomotion and adapt their locomotion pattern within an expected feasibility of a candidate learning algorithm for the biped walking by using an algorithm of two elements to learning appropriate foot placement and as estimated walking cycle timing with the application of model based reinforcement learning, in such that a model of Poincare map and the chosen control action based on a computed a value function (Jun. M et al, 2004). The gap of their study was RL problem representation which was computed with discrete state space the gaps of their study was the RL problem representation which was computed with discrete state space action within a derived model dynamic of the walking biped humanoid robot, this was not the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a mathematical interpretation of the blackbox.

The Erik et al; in their work presented the RL problem for the online autonomous reinforcement learning (Erik et al, 2010)**.** In their work, they had derived t In their work, they had derived the main hardware and software requirements that which an RL problem should fulfilled, and they had designed their own biped robot LEO, that was specifically designed to meet their requirements (Erik et al, 2010). And finally, they verified its aptitude in autonomous walking experiments using a pre-programmed controller. Although there was a room for improvement in their own design, that the robot was able to walk, fall and stand up without human intervention for the time duration of 8 hours, during which it was taken over 43; 000 footsteps (Erik et al, 2010).

In the other paper of robust biped locomotion using deep deterministic reinforcement learning on the top of analytical control approach by Mohammad et al, they had proposed a flexible framework to generate robust biped locomotion with the thighs coupling between an analytical walking approach and deep reinforcement learning (Mohammad et al, 2021). The core of the framework was a specific dynamics model which abstracts a humanoid

dynamics model into two masses for modeling upper and lower body. This dynamics model was used to design an adaptive refence trajectories planner and an optimal controller which are fully parametric (Mohammad et al, 2021). In their works they had sated the simulations are performed to validate the performance of the frameworks using an official RoboCup 3D league simulation environment. The results they had validated the performance of the performance of the framework, not only in creating a fast and stable gait but also in learning to improve the upper body efficiency (Mohammad et al, 2021).

In other concern that was presented by Akila et al; explored that the RL namely deep query network that was made a robot to walk in the controlled environment and the main objectives of their papers was to made a robot to learn to walk using a model in the hardware with six forms of movements and those the hardware that could be capable comminates for the learning model which was run on the local machine through a USB port connection (Akila et al, 2021). From the generalization they had made by them from the result they had gotten within the approach which they had used as they showed that there was a potential of physicals robots learning ways which can made robot to walk without pre-installed model and simulations, however to be able to establish it concretely they had faced many factors that have to be ensured and validated about the system (Akila et al, 2021). Also, there are several stages that lack precision and completeness in their implementation, when it comes to the hardware environment as the most of them were built with commercially available tools and materials.

Table 2.2: Summarized literature review on AI method for the biped humanoid robot

References	**Titles**	**Methodology**	**Contribution**	**Gap**
(Jun. M et al, 2004).	A simple reinforcement learining aligorthim for biped walking (Jun. M et al, 2004).	Model based RL algorithm for biped walking robot in which the robot learned appropriately for placing the wing leg (Jun. M et al, 2004)..	The desired walking cycle frequency based on online measurements which learn and adapt locomotion pattern within an explored feasibility of a candidate learning algorithm for the biped walking robot (Jun. M et al, 2004).	One of the gaps of their study was the RL problem representation was computed with discrete state space action within a derived model dynamic this was not the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a mathematical

				interpretation of the blackbox (Jun. M et al, 2004).
(Erik et al, 2010)	The design of LEO; a 2D bipedal walking robot for online autonomous einforcement learining(Erik et al, 2010).	Desecrate state action pairs hierarchical reinforcement learning Discreate state action hierarchical Reinforcement Learning (Erik et al, 2010).	They had derived the main hardware and software requirements that which and software requirements that which an RL problem should fulfilled and they designed their own biped robot LEO (Erik et al, 2010).	The agent they had used in their paper was not continuous state instead they had used the discrete markov model process [MDP]. The agent that trained with continuous state action pairs were preferable to BWR that leads to promise of AI toward General AI (Erik et al, 2010).
(Mohammad et al, 2021)	Robust biped locomotion usinf deep reinforcement learining on top of an analytical control approach (Mohammad et al, 2021)	Based on genetic algorithm (GA) and proximal policy optimization (PPO) (Mohammad et al, 2021).	The results they had validated the performance of the framework, not only in creating a fast and stable gait but also in learning to improve the upper body efficiency (Mohammad et al, 2021).	The agent they had used in their paper was not continuous state instead they had used the discrete markov model process [MDP]. The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence. The network they had used was only actor which leads to the learning agent to less autonomous (Mohammad et al, 2021).
(Diego .R & Sven.B, 2021)	DeepWalk omindirectional bipedal gait by deep reinforcement learining (Diego .R & Sven.B, 2021).	Innovative reinforcement learning (Diego .R & Sven.B, 2021).	They had achieved learning to walk by introducing a new curriculum learning method that gradually increase the task difficulty by scheduling target velocity (Diego .R & Sven.B, 2021).	The locomotion behaviors are limited to accomplished by a single control policy (Diego .R & Sven.B, 2021).

(Akila et al, 2021)	Reinforcement learing walking robot (Akila et al, 2021).	Deep query learning with neural network that was used to made a robot to walk controlled environment [DQNN] (Akila et al, 2021).	Linear model predictive control scheme which is an improvement of the original ZMP preview control scheme (Akila et al, 2021).	They had used indirect position control from torque perturbation and it is not suitable for the real-time control of RC servo motor and needs an extra tachometer sensor reading (Akila et al, 2021).

CHAPTER THREE

METHODOLOGY AND MATHEMATICAL MODELLING

3.1 Proposed Methodology

In this chapter the overall methodology and mathematical modelling of Dinkinesh biped humanoid robots was discussed, within a specified task, that to be must deliberated by Dinkinesh biped robot and the methodology of designing of those tasks oriented functionality, based on the biped locomotion and balancing strategies, path planning and the integration of the AI method to the Dinkinesh humanoid robot by considering best ways to be applied within a relevant control techniques, AI method with a highly optimized soft computing and an overall mechanical design consideration were performed with the analytic method with simulation and practical was considered to solve and verify the problem stated. So, in this thesis the 20DOF biped walking humanoid robot design was considered and the design approach comprised of its own different style of design of the mechanics, control techniques in addition with the real time implementation in servo motors, gyroscope and accelerometer, cameras and GPU boards and microcontroller accordingly in the piecewise manner. In the section below all the method were described mathematically with so as to be in considering with realtime implementation as well.

3.1.1 List of Materials required for design and implementation

3.1.1.1 List of Materials required for hardware implementation

- Raspberry Pi 4 Model B
- TFT Display for Raspberry Pi - Resistive Touch Screen
- MG996R Servo Motor
- 5MP Night Vision Camera for Raspberry Pi
- Force Sensor
- SanDisk 32GB Sdcard
- MPU6050 accelerometer + gyroscope
- Arduino mega2560
- Ultrasonic Sensor
- Infrared Sensor
- Tilt sensor module
- Microphone Mic

- Speaker

3.1.1.2 List of software used in implementation of the thesis

For the software implementation in this study, I had used python integrated development environment. Python provide an interactive programming techniques and tools form machine learning and artificial intelligence system by allowing ML and AI packages or modules inside the IDE, this includes of TensorFlow light, Kera's, OpenCV, Pandas, OpenAI, gym, NumPy, matiplot and so on.

The other good feature of python is it can be used in the hardware implementation for computer vision, image processing and it is just compact with Raspbian operation system that can be installed in the Linux based raspberry pi computer.

For the software design and analysis, I had used MATLAB2019b for the system simulation environment was done around there. Vscode studio to program a python language, C++ and this was an important tool in the design of the code suitable for the hardware programming system.

3.2 Biped locomotion and balancing strategy of humanoid robots

The biped locomotion and balancing strategies are the most important and notable concept in which all the functionality of a humanoid robot was basically depends on this sub-task of the design of the biped walker in the humanoid robot design.

The question was, what type of the biped locomotion approach and balancing strategies should be considered in the design of stable and dynamic walking biped humanoid robot? In these contexts, I also consider the ways to represent the mechanisms of the walking in the such ways of physical system constraint representation in this case, I assumed the biped walker as the inverted pendulum constraints.

However the representation of the biped walker can be represented by two legs, which mean that the IPM can be replaced by double inverted pendulum model (DIPM) by applying with a most important walking constraint through a mathematical derivation, then next by introducing a relevant control method like model predictive control to enhance the trajectory and balancing approach with the integration of center of mass projection(COM) and zero moment point (ZMP) in the efficient manner were presented with a following efficient block diagram of control process as given below in the figure 3.1.

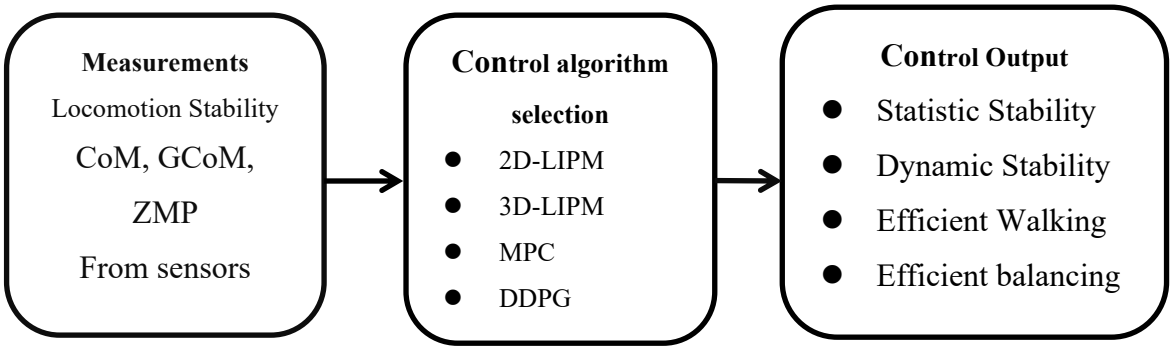


Figure 3.1: Proposed method to design locomotion and balancing strategy

As shown in the figure 3.1 above in the method to design a locomotion and balancing strategies, by assuming measurement and locomotion stability like center of mass, ground center of mass and zero moment point for the modelling of the inverted pendulum model constraints, in the two dimensional or three dimensional linear inverted pendulum model with applicability of the control method like model predictive control with the control output like static stability, dynamic stability in order to gain efficient walking and balancing output.

The relevant question out there was how to achieve this goal and how the mathematical formulation should be done for the required goal in this scope accordingly. To answer those all questions performing a flexible mathematical model was so important and the following method was described accordingly in the next section below.

3.2.1 Linear Inverted Pendulum Model

As I mentioned in the above to design the biped humanoid that can be able walk in the certain environment dynamics, I considered the double linear inverted pendulum model, but by studying a single linear inverted pendulum (LIPM) the double linear inverted pendulum can be achievable. the li

The linear inverted pendulum model is a physical model that can be used to generate biped humanoid robot walking complex set of motion. It is an extension to the cart-table and has the same equations. As in the cart-table, the LIPM simplifies the robot as a single mass point on its center of mass (CoM). It restricts the pendulum motion to the horizontal axis without any vertical movement which results in a linear state space equation. In the Figure 3.2 as provided a schematic view of the model. The linear inverted pendulum assumes that the zero-moment point (ZMP) at the origin O, which resembles to the robot's ankle, and the ankle's torque sets to zero (S. Kajita et al, 2001). In the other type of pendulum model in which it contrasted with the inverted pendulum, the cart-table model allows the ZMP to

move in the foot that is modeled by the foot contacts with the ground terrain, and the torque of ankle is not required to be zero (Shafii, 2015).

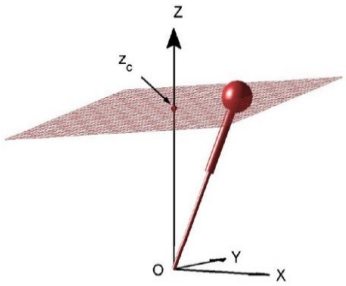


Figure 3.2: Linear Inverted Pendulum (S. Kajita et al, 2001)

However, the linear inverted pendulum model also has the drawback of not considering any angular momentum associated with the motion but it is very important to going through the position control to be simple and efficient. So, to apply the LIPM as like when a biped robot just supports its body on one leg, its dominant dynamics can be easily can be represented by a single inverted pendulum which connects the supporting foot and the center of mass of the whole robot. As the figure 3.3 depicts such an inverted pendulum consisting of a point mass and a massless telescopic leg.

So that the position of the point mass p = (x, y, z) is uniquely specified by a set of state variables q = (θ_r, $\theta_{p,\,r}$). where θ_r, indicates the angle between the pendulum and the xy-plane and θ_p indicates the angle between the pendulum and the yz-plane (Fig. 3.3). The signs of these angles are defined to fit the right-handed coordinate system, so we have $\theta_r < 0$ and $\theta_p > 0$ in the configuration of Fig. 3.3. r indicates the length of massless leg which is the distance between the origin and the point mass, meaning the distance from the hip to foot of the robot.

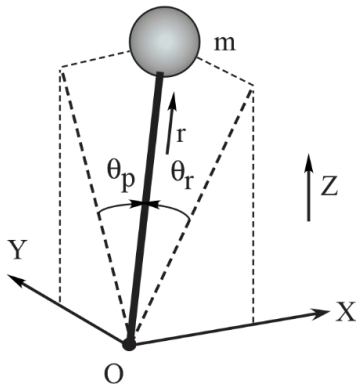


Figure 3.3: 3D Inverted Pendulum

To model the biped walking robot Interms of the LIPM constraints, let (τ_r, τ_p, f) be the actuator torque and force associated with the state variables of (θ_r, θ_p, r). With these inputs,

the equations of motion of the 3D inverted pendulum in cartesian coordinates are given by in the equation (3.1) and equation (3.2).

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r}\tau_r - mgy \tag{3.1}$$

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p}\tau_p + mgx \tag{3.2}$$

Where $C_r \equiv \cos\theta_r, C_p \equiv \cos\theta_p, D = \sqrt{C_r^2 + C_p^2 - 1}$ and m is the mass of the pendulum, and g is gravity acceleration.

And although the moving pattern of the pendulum has vast possibilities of the application, so we should have to know how to select a class of motions that would be suitable for the application of biped walking robot. For this reason, we can apply the constraints to limit the motion of the pendulum. The first constraint limits the motion in the plane with given normal vector (k_x, ky, -1) and z intersection z_c, this equation can be given below.

$$z = k_x x + k_y y + z_c \tag{3.3}$$

For a robot walking on a rugged terrain, the normal vector should match the slope of the ground and the z intersection should be the expected average distance of the center of the robot's mass from the ground. For further calculation, we can prepare the second derivatives of Eq. (3.3),

$$\ddot{z} = k_x\ddot{x} + k_y\ddot{y} \tag{3.4}$$

Substituting these constraints into Eqs. (3.1) and (3.2), we can obtain the dynamics of the pendulum under the constraints. From straightforward calculations, we can get from equation (3.1), we have the constraints to limits in which the pendulum should have the motion equation to the waking platform we introduce the term equation (3.4) and we can get the new equation (3.4) from the derivation below.

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r}\tau_r - mgy,\ \ddot{z} = k_x\ddot{x} + k_y\ddot{y} \text{ and } z = k_x x + k_y y + z_c$$

Then next we have the new equation

$$m\left(-(k_x x + k_y y + z_c)\ddot{y} + y(k_x\ddot{x} + k_y\ddot{y})\right) = \frac{D}{C_r}\tau_r - mgy$$

$$m(-k_x x\ddot{y} - k_y y\ddot{y} - z_c\ddot{y} + k_x\ddot{x}y + k_y\ddot{y}y) = \frac{D}{C_r}\tau_r - mgy$$

$$(-k_x x\ddot{y} - k_y y\ddot{y} - z_c\ddot{y} + k_x\ddot{x}y + k_y\ddot{y}y) = \frac{1}{m}u_r - gy$$

$$(-k_x x\ddot{y} - z_c\ddot{y} + k_x\ddot{x}y) = \frac{1}{m}u_r - gy = -z_c\ddot{y} + k_x(y\ddot{x} - x\ddot{y}) = \frac{1}{m}u_r - gy$$

$$z_c\ddot{y} = gy + k_x(y\ddot{x} - x\ddot{y}) - \frac{1}{m}u_r$$

$$\ddot{y} = \frac{g}{z_c} y + \frac{k_x}{z_c}\left[y\ddot{x} - x\ddot{y}\right] - \frac{1}{mz_c} u_r \tag{3.5}$$

In similar manner from equation (3.2), we have the constraints to limits, in which the pendulum should have the motion equation to the waking platform, so it is desirable to introduce a new term in equation (3.4) and we can get the new equation (3.6) by derivation the in the section below

$$m\left(z\ddot{x} - x\ddot{z}\right) = \frac{D}{C_p}\tau_p + mgx \text{ , } \ddot{z} = k_x\ddot{x} + k_y\ddot{y} \text{ and } z = k_x x + k_y y + z_c$$

$$m\left(\left(k_x x + k_y y + z_c\right)\ddot{x} - x\left(k_x\ddot{x} + k_y\ddot{y}\right)\right) = \frac{D}{C_p}\tau_p + mgx$$

$$\left(k_x x\ddot{x} + k_y y\ddot{x} + z_c\ddot{x} - k_x\ddot{x}x - k_y\ddot{y}x\right) = \frac{D}{mC_p}\tau_p + gx$$

$$\left(k_y y\ddot{x} + z_c\ddot{x} - k_y\ddot{y}x\right) = \frac{1}{m}u_p + gx = \left(\frac{k_y}{z_c}\left(y\ddot{x} - \ddot{y}x\right) + \ddot{x}\right) = \frac{1}{mz_c}u_p + \frac{gx}{z_c}$$

$$\ddot{x} = \frac{g}{z_c}x - \frac{k_y}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c}u_p \tag{3.6}$$

where u_r, u_p are new virtual inputs which are introduced to compensate input nonlinearity,

$$u_r = \frac{D}{C_r}\tau_r \tag{3.7}$$

$$u_p = \frac{D}{C_p}\tau_p \tag{3.8}$$

In the case of the robots walking on a flat plane, we can set the horizontal constraint plane with the constants of ($k_x = 0$, $ky = 0$) and now we can obtain

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{mz_c} u_r \tag{3.9}$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{mz_c} u_p \tag{3.10}$$

In the case of the biped robot walking on a slope or stairs where $k_x, k_y = 0$, we should have to find another constraint. From $x\times$ (3.5) $-y\times$ (3.6) we can obtain the following dynamic equation as shown below.

$$x\left[\ddot{y}\right] - y\left[\ddot{x}\right] = x\left[\frac{g}{z_c}y + \frac{k_x}{z_c}\left[y\ddot{x} - x\ddot{y}\right] - \frac{1}{mz_c}u_r\right] - y\left[\frac{g}{z_c}x - \frac{k_y}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c}u_p\right]$$

$$x\left[\ddot{y}\right] - y\left[\ddot{x}\right] = \left[\frac{g}{z_c}yx + \frac{xk_x}{z_c}\left[y\ddot{x} - x\ddot{y}\right] - \frac{x}{mz_c}u_r\right] - \left[\frac{g}{z_c}yx - \frac{yk_y}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{y}{mz_c}u_p\right]$$

$$x[\ddot{y}]-y[\ddot{x}]=\left[\frac{g}{z_c}yx-\frac{g}{z_c}yx+\frac{xk_x}{z_c}[y\ddot{x}-x\ddot{y}]+\frac{yk_y}{z_c}(x\ddot{y}-\ddot{x}y)-\frac{x}{mz_c}u_r-\frac{y}{mz_c}u_p\right]$$

$$x[\ddot{y}]-y[\ddot{x}]=\left[-\frac{x}{mz_c}u_r-\frac{y}{mz_c}u_p\right]$$

$$x\ddot{y}-\ddot{x}y=\frac{-1}{mz}\left(u_r x+u_p y\right) \tag{3.11}$$

Therefore, we have the same dynamics of Eq. (3.9) and Eq. (3.10) in the case of an inclined constraint plane when the following new constraint is introduced about the inputs as,

$$u_r x+u_p y=0 \tag{3.12}$$

This also implies that the ankle torques should be given in a manner of not generating the yaw moment so that the *Z* element of angular momentum is conserved. Eqs. (3.9) and (3.13) are independent linear equations. From the two equation the only parameter which governs those dynamics is z_c, i.e., the z intersection of the constraint plane and the inclination of the plane never affects the horizontal motions. Note that the original dynamics were nonlinear and we have derived linear dynamics without using any approximation. So now let us to call this the three-dimensional linear inverted pendulum mode (3D-LIPM). From the nature of the 3D-LIPM. So, we can examine a nature of trajectories under the 3D-LIPM with zero input torques ($u_r = u_p = 0$).

$$\ddot{y}=\frac{g}{z_c}y \tag{3.13}$$

$$\ddot{x}=\frac{g}{z_c}x \tag{3.14}$$

With a given initial condition, these equations determine trajectories in 3D space. Figure 3.12 shows two examples.

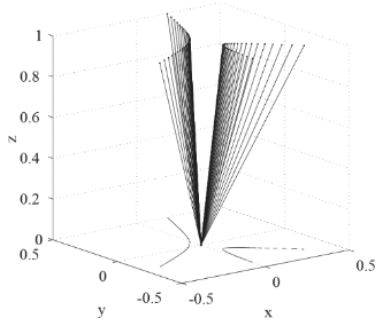


Figure 3.4: 3D Linear Inverted Pendulum Mode

Eqs. (3.13) and (3.14) can be regarded as a repulsive force field for a unit mass.

$$f = \frac{g}{z_c} r \tag{3.15}$$

A unit mass gets a force of magnitude *f* which is proportional to the distance *r* between the mass and the origin. Therefore, the 3D-LIPM with zero input torque can be considered as dynamics under the central force field. Figure 3.5 shows a 3D-LIPM trajectory which is projected onto *xy* plane. Motion along *y* and *x* is governed by Eqs. (3.13) and (3.14) respectively. In Fig.3.5, it is also shown another coordinate frame x'y' which rotates θ from the original frame *xy*. Since the 3DLIPM is a dynamic under the central force field, the motion along *y* and *x* is also governed by the identical equations with Eqs. (3.13) and (3.14). This gives us great advantages in generating walking pattern as we will see in the following sections.

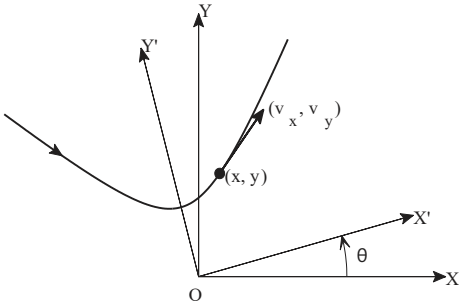


Figure 3.5: 3D-LIPM projected onto xy plane

The pattern generation along a local axis

In the pattern generation along the local axis, the problem becomes a control of the motion along x or *y*-axis for each step. Let us assume that the robot is repeating single support phase of duration T_s and double support phase of duration T_{dbl}. Figure 3.6 illustrates successive steps in *X*-direction. The initial body state $\left(x_i^{(n)}, v_i^{(n)}\right)$and the final body state $\left(x_f^{(n)}, v_f^{(n)}\right)$ have the relationship given by the equation below.

$$\begin{pmatrix} x_f^{(n)} \\ v_f^{(n)} \end{pmatrix} = \begin{pmatrix} C_T & T_c S_T \\ \frac{S_T}{T_C} & C_T \end{pmatrix} \begin{pmatrix} x_i^{(n)} \\ v_i^{(n)} \end{pmatrix} \tag{3.16}$$

Where $C_T \equiv cosh(T_S/T_C)$, $S_T \equiv sinh(T_S/T_C)$, $T_c \equiv \sqrt{z_c/g}$

To control the walking speed, we must change the foothold (point E) to modify the initial condition of the support phase (D'→F). When the desired status at the end of support (point F) was given as (x_d, v_d), we can define the norm of the error with certain weight *a, b* > 0 as $N \equiv a\left(x_d - x_f^{(2)}\right)^2 + b\left(v_d - v_f^{(2)}\right)^2$.

By substituting Eq. (3.16) into this definition and calculating the foothold of x_i which minimizes N, we can obtain a proper control law,

$$x_i^{(2)} = \left(aC_T \left(x_d - S_T T_c v_d \right) + b\, S_T / T_c \left(v_d - C_T v_f \right) \right) / D_T \tag{3.17}$$

Where $D_T \equiv aC_r^2 + b\left(S_T / T_c \right)^2$

To determine the foothold E, we also need the distance that body travels in the double support. The distance d is given by

$$d = v_f^{(1)} T_{dbl} \tag{3.18}$$

The motion of the swing leg was planned to arrive at the point E at the expected touchdown time (dashed curve from A to E in Fig. 3.6).

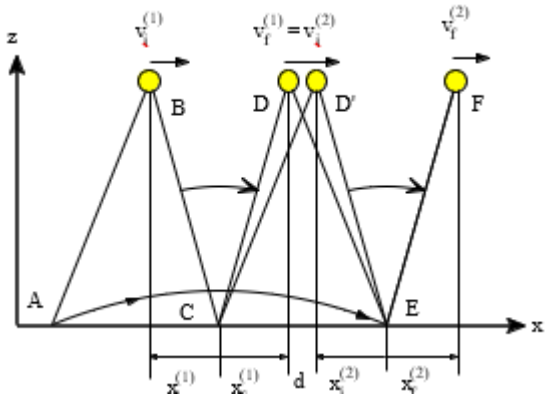


Figure 3.6: LIPM two successive steps in the sagittal plane pattern generation.

The body travels from B to D in the single-leg support phase, then moves from D to D' in the double-leg support phase with constant speed $v_f^{(1)}$, and then travels D' to F in the second single-support phase. While the body moves from B to D, the tip of the swing leg travels from A to E (dashed curved line). By changing the position of E, we can control the final body speed $v_f^{(2)}$ at the point F. So, in the above modeling of the LIPM for the biped walking robot representation by the relevant model assumption with ZMP output and COM trajectory, now for the design of Dinkinesh biped walking humanoid in the aspects of double linear inverted pendulum, I should have to design the kinematics structure and mechanical body proportion design accordingly in the section below.

3.2.1.1 Model parameters and kinematics design

In the mechanics and motion planning, the fundamental background of motion manipulation and control techniques fundamentally relies on the kinematics structure and the frame assignment of the mechanical bodies. In general, conventionally there are two basic

kinematics modeling of mechanical system, one is determining how the given mechanical system should represented in the predetermined position and orientation of the system that meet the task domain by the end effectors from all the given initial angles of each joint that's maybe either prismatic or revolute, this techniques of computing the end effectors position and orientation from the initial joint angles so called the forward kinematics.

And the other important in dynamic motion control is inverse kinematics this is also the reverse of forward kinematics in which from the given mechanical system position and orientation of the end effectors, the final controllable joint angle is computed, one of the premise of inverse kinematics is that can simplify the coordinate stepping of joint position allow the inverse angle of the joint to be easily controllable to the final control variable and also simplify the control mechanisms of the mechanical system based on the position control of the joint space. So, it's obvious that modelling of a mechanical system, the first step is to model the kinematics of the system.

Before going to design and model the locomotion and balancing strategy, Formulating the kinematics problem derivation is very crucial to maintain efficient dynamics went to in through successful preprogrammed hardware control. Because of the kinematic problem formulation is one of the most notable and important consideration in designing robotics system, in this regard the overall controllable kinematics angle of the workspace of Dinkinesh humanoid robot contained of twenty degrees of freedom (20DOF), with each leg has six DOF and totally the two-leg contained of 12 DOF, each hand contained of three DOF and totally both hands contained of six DOF and finally with neck two DOF. Each actuator joint connected by the links, which connect the body of robot, this link contains seven links below the torso with both legs and fives links above the torso and totally Dinkinesh mechanically linked with twelve links entirely for the full body design and kinematics calculation.

The overall body posture of Dinkinesh shown below in the figure (3.7). As shown in the figure (3.7), Dinkinesh humanoid robot contain 20 DOF, in the section below the researcher going to design the overall kinematics problem, to formulate the dynamic motion control problem modelling, there are two techniques; the one is based on direct mathematical interpolation method which utilized based on polynomials and the spline functions in which the robots perform the well-known motion pattern from the geometrical relation.

The second method based on Denaviet-Hertainberg (DH) Parameters. The first method is just feasible in the trajectory generation and specified motion pattern; however, it takes higher level of computation tasks this is because of the humanoid robot should be capable to perform several versatile motion patterns and sometimes it should be addresses external unpredictable perturbation, this is a special requirement of the control techniques needed. In this study both methods used in modelling based on the specified requirements of the motion parameters.

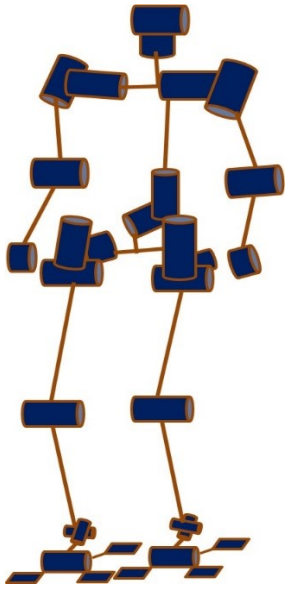


Figure 3.7: The overall body posture of Dinkinesh Humanoid

In order to design the mechanical system of the humanoid robot, it is essential to consider the overall dimension of the body and the total height of the robots. Dinkinesh humanoid is 1.2meter(120cm), the height of the humanoid robots can determine every size and width of the other body parts, this was aspired from the natural human body experiments just mentioned in Appendix B.

So, the figure that described in the figure 3.8 was modelled based on the scientific human body ratio index obtained from the john Hopkins university of medical research institute that provided in the appendix B.

And the design requirements in this study derived from mentioned in the figure (3.8). To design the biped humanoid robot by deriving the inverse kinematics I had used the closed form of inverse kinematics for humanoid robot by ali eat al, in which by using Denaviet-Hertainberg method by applying the inverse transform techniques.

DH kinematics modeling of 12DOF biped humanoid

In order to describe the link and joint configuration of a robotic system, Denavit Hartenberg (DH) parameters are often used. These parameters describe rotation and translation between the frames of the system.

There are two different forms of DH parameters, classic and modified DH-parameters. The difference between classic-DH and modified-DH parameters are the locations of the coordinates system attachment to the links and the order of the performed transformations. This difference was also shown in equation 3.18.

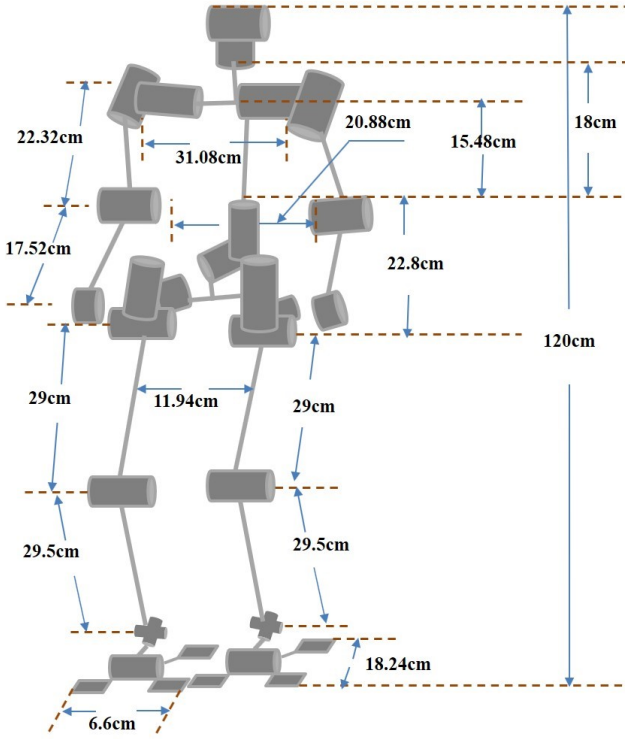


Figure 3.8: The Dinkinesh humanoid Body Dimension Height proportion

Denavit and Hartenberg have proposed a way of creating a homogeneous transformation matrix that describes the position and orientation of the i^{th}coordinate frame, which is attached to the joint at one end of the link, in the $(i-1)^{th}$coordinate frame that is fixed to the joint at the other end of the link, as a function of the joint state. According to DH they concluded that this transformation matrix can be fully described using only four parameters, known as Denavit-Hartenberg (D-H) parameters: a_i, α_i, d_i and θ_i. Before we explain these parameters, we first establish the reference frame of each joint with respect to the reference frame of its previous joint (refer to Figure 3.9):

- The origin is located at the intersection of the common perpendicular to axes $z_{i-1}$and z_i, and joint axis z_i.

- The z_i-axis is set to the direction of the joint axis.
- The x_i-axis lies along the common perpendicular to axes $z_{i-1}$and z_i, and is oriented from z_{i-1} to z_i.
- The y_i-axis follows from the x_i and z_i axes to form a right-handed coordinate system. The D-H parameters are described as follows:
- a_i: the length of the common perpendicular to axes z_{i-1} and z_i. ai length is also known as link length.

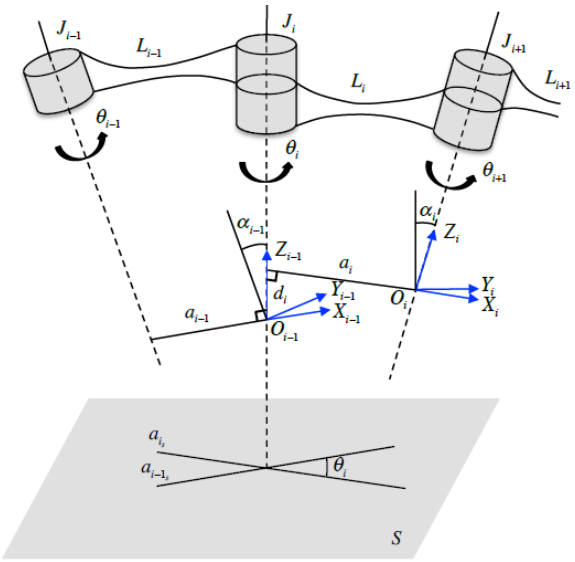


Figure 3.9 The Denaviet-Hartenberg Kinematics Frame

- α_i : the angle around xi that makes the vector z_i codirectional with the vector z_{i-1}. The angle is also called link twist.
- d_i: the algebraic distance along axis z_{i-1} to the point where the common perpendicular to axis z_i is located. In the bibliography this parameter is also called link offset.
- θ_i: the angle around z_i that makes the common perpendicular codirectional with the vector x_{i-1}. The angle around z_i is called joint angle.

Now, we can move from the base reference frame of a certain joint to the transformed reference frame of this joint using the transformation matrix ${}^{i-1}T_i$, which consists of two translations and two rotations parametrized by the D-H parameters of the joint:

$$ {}^{i-1}T_i = Rot(x,\alpha_i)\,Trans(\alpha_i,0,0)\,Trans(0,0,d_i) \tag{3.19} $$

The analytical form of the resulting matrix from the above composition is just as shown below in the equation (3.20) as the following.

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i \\ \sin\theta_i\cos\alpha_i & \cos\theta_i\cos\alpha_i & -\sin\alpha_i & -d_i\sin\alpha_i \\ \sin\theta_i\sin\alpha_i & \sin\theta_i\sin\alpha_i & \cos\alpha_i & d_i\cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20} $$

The transformation matrix ${}^{i-1}T_i$ can be also be formulated as

$$ {}^{i-1}T_i = Rot(z,\theta_i)\,Trans(0,0,d_i)\,Trans(a_i,0,0)\,Rot(x,\alpha_i) \tag{3.21} $$

Then the corresponding analytical form of the D-H matrix is expressed as follows

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & -a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22} $$

I had used the Denavit-Hartenberg (D-H) matrix representation for each link to describe the rotational and translational relationship between adjacent links. Using the D-H matrix representation and following the link coordinate frame assignment can be outlined simply with DH algorithm, one can assign link coordinate frames that are consistent with the positive direction of rotation of joints of a humanoid robot.

First, I establish two base coordinate frames $B_1$and B_2 for the Dinkinesh humanoid robot. B_1 is established at the center of the neck and is the reference coordinate frame for the arms and the head, and B_2 is the reference coordinate frame for the legs. B_2 is linked to B_1 through a waist joint, and there is a simple link transformation matrix between B_1 and B_2. As a result,B_1 is considered to be the global base coordinate frame for the whole robot.

Since the general kinematic structures of the left arm/leg of a Dinkinesh humanoid robot are identical to those of the right arm/leg, we have assigned identical coordinate frames for the left and right limbs. Figures 3.10 and 3.11 show the assigned link coordinate frames and their D-H parameters for the right arm and the right leg, respectively.

From the established link coordinate frames and the D-H parameters, the position and orientation of the end-effector of a limb can be obtained by chain-multiplying the 6 link-transformation matrices together to obtain the spatial displacement of the 6th coordinate frame with respect to the base/reference coordinate frame:

$$ {}^{0}T_6 = \prod_{i=1}^{6} {}^{i-1}A_i = {}^{0}A_1\,{}^{1}A_2\,{}^{2}A_3\,{}^{3}A_4\,{}^{4}A_5\,{}^{5}A_6 $$

$$= \begin{bmatrix} x_6 & y_6 & z_6 & p_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

where x_i, y_i, and z_i represent the unit vectors along the principal axes of the coordinate frame i, ${}^{i-1}A_i$ is a general link transformation matrix, relating the ith coordinate frame to the $(i-1)^{th}$coordinate frame, and [n; s; a; p] represents the normal vector, the sliding vector, the approach vector, and the position vector of the hand, respectively.

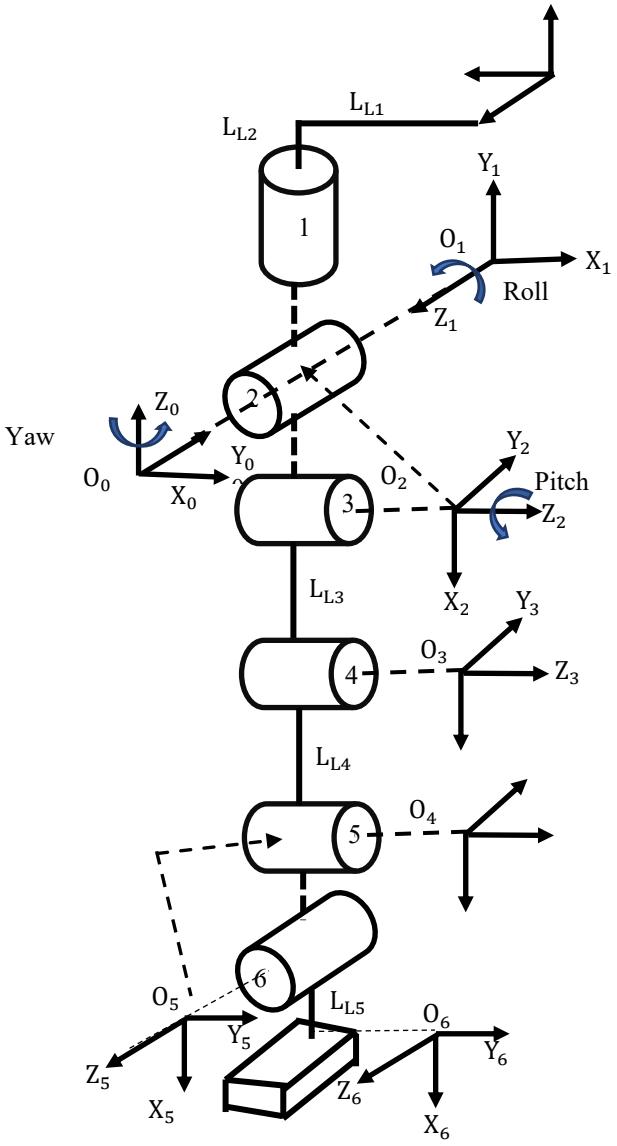


Figure 3.10: Link Coordinate Frames of the Right Leg of a Dinkinesh humanoid robot and its D-H Parameters

This forward kinematic equation will be used in deriving the closed-form joint solution for a Dinkinesh robot. For the given the desired position and orientation of the end effector of a limb (an arm or a leg) and the geometric link parameters with respect to a reference coordinate system, the inverse kinematic position problem is to find a closed form joint

solution for positioning the end-effector with the desired position and orientation. And if a such closed-form joint solution exists, then it is desirable and possible to determine how many different joint solutions will satisfy the same condition. If a limb of the robot has six joints, we should, in principle, be able to find a closed-form joint solution. A joint solution is said to be "closed-form" if the unknown joint angles can be solved for symbolically in terms of the arc-tangent function. Hence, one should be able to find closed-form solutions for these robots, and since the kinematic configuration of all these humanoid robots is almost the same as a Dinkinesh robot, the developed closed-form joint solution will be applicable to these humanoid robots with only slight modifications.

In tackling the inverse kinematic position problem, Pieper indicated a closed-form joint solution exists if a robot manipulator's three adjacent joint axes are parallel to one another or they intersect at a single point. by considering a limb in which the joint axes of the last three joints intersect at a point, the position vector **p** to the wrist coordinate frame decouples the limb into positioning and orientation subsystems, and obviously it is a function of just the first three joint angles; the last three joint angles contribute nothing to the position but the orientation. Hence, there are three equations with known variables p_x, p_y and p_z and three unknown joint angles θ_1, θ_2 and θ_3.

The last three joint angles are determined from [n, s, a] and the solved first three joint angles. The key concept here is that the robot is decoupled with the three of the six joint angles for positioning and three joint angles for orientation. Unfortunately, for the limbs of the existing humanoid robots, the joint axes of the last three joints do not intersect at a point; for example, the last three joint axes of the right leg/arm of a Dinkinesh humanoid robot do not intersect at a point. As a result, the position vector **p** is a function of four joint angles θ_1, θ_2, θ_3, and θ_4. Hence, obtaining a closed-form joint solution becomes complicated, if not impossible. However, a closer examination reveals that the joint axes of the first three joints do intersect at a point. This means that if we view the IK-problem with the joint angles in reverse order; that is, with the position/orientation of the base coordinate frame referenced to the end-effector coordinate frame, the new position vector, denoted as $\mathbf{p}^0$, is a function of only the three joint angles θ_4, θ_5 and θ_6. This new position vector $\mathbf{p}^0$ decouples the limb into positioning and orientation subsystems. To solve the IK-problem in this reverse way, we take the inverse of both sides of Eq. (3.23) so that the new composite link transformation matrix $\mathbf{T}^0$ is now referenced to the end-effector coordinate frame; that is,

$$T' = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} n' & s' & a' & p' \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{6}A_5\,{}^{5}A_4\,{}^{4}A_3\,{}^{3}A_2\,{}^{2}A_1\,{}^{1}A_0 = {}^{6}A_0 \tag{3.24}$$

Thus, the novelty here is to observe the intersection of 3 adjacent joint axes in the kinematic chain for decoupling the robotic arm/leg system into positioning and orientation subsystems for solving its joint solution.

A. Joint Solution for the Right Leg of a Dinkinesh humanoid Robot

The transformation matrix from the base coordinate frame B_2 attached to the waist to the first coordinate frame of the right leg is

$${}^{B2}A_0 = \begin{bmatrix} -1 & 0 & 0 & L_{L1} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -L_{L2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

To obtain the solution for the last three joint angles θ_4, θ_5 and θ_6, we equate the elements of the position vector $\mathbf{p}^0$ in both sides of Eq. (3.24) as

$$-C_6\left(C_{45}L_{L3} + C_5L_{L4}\right) = p_x' + L_{L5} \tag{3.26}$$

$$-S_6\left(C_{45}L_{L3} + C_5L_{L4}\right) = p_y' \tag{3.27}$$

$$-S_{45}L_{L3} - S_5L_{L4} = p_z' \tag{3.28}$$

Were $S_i \equiv sin\,\theta_i$, $C_i \equiv cos\,\theta_i$, $S_{ij} \equiv sin(\theta_i + \theta_j)$, $C_{ij} \equiv cos(\theta_i + \theta_j)$, and L_{li} are the geometric link parameters in fig 3.10 such that it is evident those three equations have three unknown θ_4, θ_5 and θ_6. By squaring and adding these three equations, we can obtain C_4 and then S_4 from C_4, and from which we can solve for the joint solution θ_4,

$$C_4 = \frac{\left(p_x' + L_{L5}\right)^2 + p_y'^2 + p_z'^2 - L_{L3}^2 - L_{L4}^2}{2L_{L3}L_{L4}}$$

$$\theta_4 = a\tan 2\left(\pm\sqrt{1 - C_4^2}, C_4\right) \tag{3.29}$$

Were such that atan2(y, x) is an arc-tangent function, which returns $tan^{-1}\left(\frac{y}{x}\right)$ adjusted to the proper quadrant. By using Eqs. (3.26) and Eqs. (3.27), adding and expanding them, we get Eqs. (3.30),

$$C_4\left(C_4L_{L3} + L_{L4}\right) - S_4S_5L_{L3} = \pm\sqrt{\left(p_x' + L_{L5}\right)^2 + p_y'^2} \tag{3.30}$$

By expanding Eqs. (3.30), we obtain

$$S_4\left(C_4L_{L3}+L_{L4}\right)-C_5\left(S_5L_{L3}\right)=-p_z^{'} \tag{3.31}$$

Let that $C_4L_{L3}+L_{L4}=rC_\psi$ and $S_4L_{L3}=rS_\psi$ and by substituting them in to Eqs. (3.30) and (3.31), we got the corresponding equation (3.32).

$$rC_{5\psi}=\pm\sqrt{(p_x^{'}+L_{L5})^2+p_y^{'2}} \tag{3.32}$$

$$rS_{5\psi}=-p_z^{'} \tag{3.33}$$

Where $r=\pm\sqrt{(p_x^{'}+L_{L5})^2+p_y^{'2}+p_z^{'2}}$ and $\psi=a\tan 2\left(S_4L_{L3},C_4L_{L3}+L_{L4}\right)$ Dividing Eq. (3.30) by Eq. (3.29), we obtain $\tan\left(\theta_5+\psi\right)$, which finally gives the joint solution for θ_5

$$\theta_5=a\tan 2\left(-p_z^{'}\pm\sqrt{\left(p_x^{'}+L_{L5}\right)^2+p_z^{'2}}\right)-\psi \tag{3.34}$$

Dividing Eq. (3.27) by Eq. (3.26), we get the joint solution for θ_6

$$\theta_6=a\tan 2\left(p_y^{'},-p_x^{'}-L_{L5}\right) \tag{3.35}$$

And if $C_{45}L_{L3}+C_5L_{L4}<0$then we have $\theta_6=\theta_6+\pi$. In order to obtain the three joint angles θ_1, $\theta_2$and θ_3, we can use the inverse transform method by moving the link transformation matrix 6A5 to the left-hand side of Eqs. (3.24). This results in a matrix equation that I had labeled as G2 equation.

$$G_{2-leg}^{(LHS)}={}^5A_6\begin{bmatrix} n^{'} & s^{'} & a^{'} & p^{'} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$=\begin{bmatrix} C_6n_x^{'}-S_6n_y^{'} & C_6s_x^{'}-S_6s_y^{'} & C_6a_x^{'}-S_6a_y^{'} & C_6p_x^{'}-S_6p_y^{'}+C_6L_{L5} \\ S_6n_x^{'}+C_6n_y^{'} & S_6s_x^{'}+C_6s_y^{'} & S_6a_x^{'}+C_6a_y^{'} & S_6p_x^{'}+C_6p_y^{'}+S_6L_{L5} \\ n_z^{'} & s_z^{'} & a_z^{'} & p_z^{'} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And also the right-hand side of G2 is just act like as follows,

$$G_{2-leg}^{(LHS)}={}^5A_0={}^5A_4\,{}^4A_3\,{}^3A_2\,{}^2A_1\,{}^1A_0$$

$$= \begin{bmatrix} C_1C_2C_{345} - S_1S_{345} & S_1C_2C_{345} + C_1S_{345} & S_2C_{345} & -C_{45}L_{L3} - C_5L_{L4} \\ -C_1S_2 & -S_1S_2 & C_2 & 0 \\ C_1C_2S_{345} - S_1C_{345} & S_1C_2S_{345} - C_1C_{345} & S_2S_{345} & -S_{45}L_{L3} - S_5L_{L4} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By comparing the elements (2,3) of the LHS and RHS of G2, we can find C_2 and S_2, and from which we can obtain the joint solution for θ_2.

$$\theta_2 = a\,tan\,2\left(\pm\sqrt{1-\left(S_6a_x^{'} + C_6a_y^{'}\right)^2}, S_6a_x^{'} + C_6a_y^{'}\right) \tag{3.36}$$

By comparing the elements (2,1) and (2,2) of the LHS and RHS of G2, we get two equations from these two elements,

$$S_1S_2 = -S_6s_x^{'} - C_6s_y^{'} \text{ and } C_1S_2 = -S_6n_x^{'} - C_6n_y^{'}$$

Then by dividing those two equations, we obtain the joint solution for θ_1,

$$\theta_1 = a\tan 2\left(-S_6s_x^{'} - C_6s_y^{'}, -S_6n_x^{'} - C_6n_y^{'}\right) \tag{3.37}$$

And if $S_2 < 0$ then $\theta_1 = \theta_1 + \pi$.

By the comparison laid on the elements (1, 3) and (3,3) of the LHS and RHS, we get the two equations in the S_{345} and C_{345}, and by dividing these two equations, we find the joint angles θ_{345}.

$$S_2S_{345} = a_z^{'} \text{ and } S_2C_{345} = C_6a_x^{'} - S_6a_y^{'}$$

$$\theta_{345} = a\tan 2\left(a_z^{'}, C_6a_x^{'} - S_6a_y^{'}\right) \tag{3.38}$$

And if $S_2 < 0$ then we have $\theta_{345} = \theta_{345} + \pi$. Finally, from the equation Eq. (3.39), we can find the joint solutions θ_3.

$$\theta_3 = \theta_{345} - \theta_4 - \theta_5 \tag{3.39}$$

The above closed-form joint solution can be easily applied to the left leg by replacing $+L_{L1}$ with $-L_{L1}$ in Eq. (3.25). we should mention that for the legs, singularity due two collinear joint axes do not exist within the valid joint-angle constraints.

B. Joint Solution for the Right Arm of a Dinkinesh humanoid Robot

The transformation matrix from the base coordinate frame B1 attached to the neck to the first coordinate frame of the right arm is,

$$ {}^{B1}A_0 = \begin{bmatrix} 0 & 0 & 1 & L_{A1} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.40} $$

Next, we obtain G2 for the right arm using the inverse transform method,

$$ G_{2-arm}^{(LHS)} = \begin{bmatrix} g_{211} & g_{212} & g_{213} & C_6\left(p_x^{'} + l_{A4}\right) - S_6 p_y^{'} \\ g_{221} & g_{222} & g_{223} & S_6\left(p_x^{'} + l_{A4}\right) + C_6 p_y^{'} \\ g_{231} & g_{232} & g_{233} & p_z^{'} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ G_{2-arm}^{(LHS)} = \begin{bmatrix} g_{211} & g_{212} & g_{213} & S_4 C_5 l_{A2} \\ g_{221} & g_{222} & g_{223} & -C_4 l_{A2} - l_{A3} \\ g_{231} & g_{232} & g_{233} & S_4 S_5 l_{A2} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

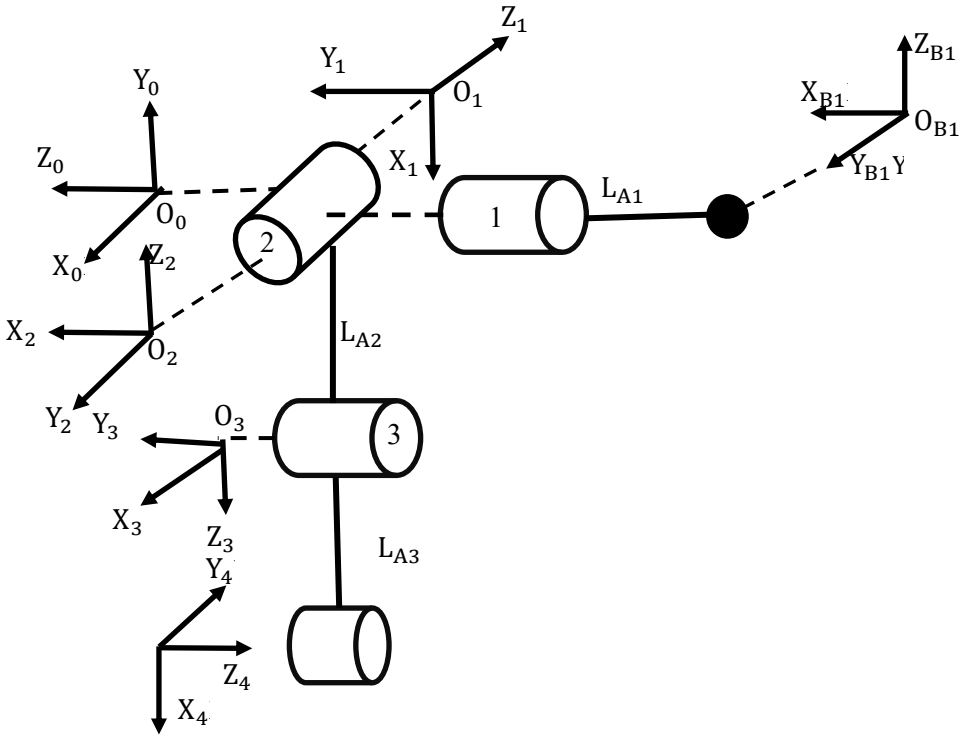


Figure 3.11 Link Coordinate Frames of the Right Arm of a Dinkinesh humanoid Robot and its D-H Parameters.

By comparing the elements (1,4), (2,4) and (3,4) of the LHS and RHS of G2, we have

$$ C_6\left(p_x^{'} + L_{A4}\right) - S_6 p_y^{'} = S_4 C_5 L_{A2} \tag{3.41} $$

$$S_6\left(p_x^{'}+L_{A4}\right)+C_6p_y^{'}=-C_4L_{A2}-L_{A3} \tag{3.42}$$

$$p_z^{'}=S_4S_5L_{A2} \tag{3.43}$$

Let $p_x^{'}+L_{A4}=rC_\psi$ and $p_y^{'}=rS_\psi$, and substitute them into equation (3.41), (3.42) and (3.43), we get, correspondingly,

$$rC_{6\psi}=S_4C_5L_{A2} \tag{3.44}$$

$$rS_{6\psi}=-C_4L_{A2}-L_{A3} \tag{3.45}$$

$$p_z^{'}=S_4S_5L_{A2} \tag{3.46}$$

Where $\sqrt{\left(p_x^{'}+L_{A4}\right)^2+\left(p_y^{'}\right)^2}$ and $\psi=a\tan 2\left(p_y^{'},p_x^{'}+L_{A4}\right)$ by squaring equation (3.44), (2.45) and (2.46) and adding them up, we can obtain C_4 and then S_4, and from which we can find the joint solution for θ_4

$$C_4=\frac{\left(p_x^{'}+L_{A4}\right)^2+\left(p_y^{'}\right)^2+\left(p_z^{'}\right)^2-l_{A2}^2-l_{A3}^2}{2l_{A2}l_{A3}}$$

$$\theta_4=a\tan 2\left(\pm\sqrt{1-\left(C_4\right)^2},C_4\right) \tag{3.47}$$

From the equation (3.46), we can find S_5 and then C_5,and from which we can find the joint solution θ_5,

$$S_5=\frac{p_z^{'}}{\left(S_4l_{A4}\right)};\theta_5=a\tan 2\left(S_5,\pm\sqrt{1-S_5^2}\right) \tag{3.48}$$

By dividing equation (3.46). we get

$$\tan\left(6\psi\right)=\tan\left(\theta_6+\psi\right)=\frac{S_{6\psi}}{C_{6\psi}}=\frac{C_4l_{A2}-l_{A3}}{S_4C_5l_{A2}} \tag{3.49}$$

And from which we obtain the joint solution θ_6

$$\theta_6=a\tan 2\left(-\left(C_4l_{A2}+l_{A3}\right),S_4C_5l_{A2}\right)-\psi \tag{3.50}$$

For the solution of the rest of the joint, we can use G4,

$$G_{4-arm}^{(LHS)} = \begin{bmatrix} C_1C_2C_3 - S_1S_3 & S_1S_3 + S_1C_2C_3 & S_2C_3 & 0 \\ -C_1S_2 & -S_1S_2 & C_2 & l_{A2} \\ S_1C_3 + C_1C_2S_3 & S_1C_2S_3 - C_1C_3 & S_2S_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G_{4-arm}^{(LHS)} = \begin{bmatrix} g_{411} & g_{412} & g_{413} & g_{414} \\ g_{421} & g_{422} & g_{423} & g_{424} \\ g_{431} & g_{432} & g_{433} & g_{434} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By comparing the elements (1.3) of the LHS of G4, we can get C2 and then S2, and the joint solution θ_2 from them,

$$C_2 = g_{423} = a_z^{'}S_4S_5 - a_y^{'}(C_4C_6 + S_4C_5S_6) - a_x^{'}(C_4S_6 - S_4C_5C_6)$$

$$\theta_2 = a\tan 2\left(\pm\sqrt{1-C_2^2}, C_2\right) \tag{3.51}$$

By comparing the elements (1,3) and (3,3) of LHS and RHS of G4, we can get two equations. By dividing these two equations, we can find the joint solution θ_3

$$g_{413} = a_x^{'}(C_4C_5C_6 + S_4S_6) + C_4S_5a_z^{'} + a_y^{'}(S_4C_6 - C_4C_5S_6)$$

$$g_{433} = S_5C_6a_x^{'} - C_5a_z^{'} - S_5s_6a_y^{'}$$

$$\theta_3 = a\tan 2\left(g_{433}, g_{413}\right) \tag{3.52}$$

And if $S_2 < 0$ and then $\theta_3 = \theta_3 + \pi$. By comparing the elements (2,1) and (2,2) of LHS and RHS of G4, we get two equations. By dividing these two equations, we can find the joint angle θ_1,

$$g_{421} = n_x^{'}(S_4C_5C_6 - C_4S_6) + S_4S_5n_z^{'} - n_y^{'}(S_4C_5S_6 + C_4C_6)$$

$$g_{422} = s_x^{'}(S_4C_5C_6 - C_4S_6) + S_4S_5s_z^{'} - s_y^{'}(S_4C_5S_6 + C_4C_6)$$

$$\theta_1 = a\,tan\,2\,(-g_{422}, g_{421}) \tag{3.53}$$

And if $S_2 < 0$, then $\theta_1 = \theta_1 + \pi$.

The closed-form joint solution can be applied to the left arm with $-l_{A1}$ instead of $+l_{A1}$ in equation (3.40). The joint solution for the right legs and arm of a Dinkinesh humanoid robot, I observed that each of the θ_2,θ_4 and θ_5 have two solutions, which yielding a total of 8 possible solution**.**

3.2.1.2 Kinematics Modeling of Walking Robot in Flat Surface from Geometrical Relation

It is also possible to exploit an effective technique for walking of biped trajectory generation by means of inverse kinematics problem, by apply the polynomials interpolation like qubit spline functions for the reason of that these are so good functions for specifying trajectory of each joint in the robot, since splines are differentiable and easy for evaluating.

Respecting to the break points (i.e., step length and geometrical conditions), the ankle trajectory can be generated. By using $n+1$, different points are becoming more suitable to calculate coefficients of a polynomial in order of n and as a result, this polynomial would be unique. The lengths of links are known. Therefore, the position of links during walking can be found by applying the geometrical relations. It is necessary to introduce some of useful parameters and assumptions in modeling of Dinkinesh biped robot. These parameters are hip and foot parameters. When the robots walk in sagittal plane hip parameters include vertical and horizontal displacements (z_h, x_h), which are shown in *Fig. 3.12*. These parameters are used in the trajectory of the hip.

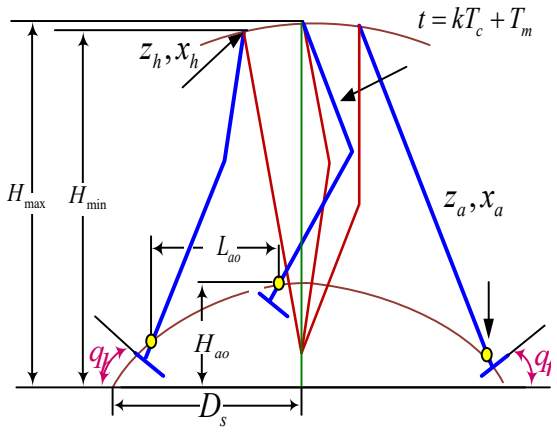


Figure 3.12: Biped robot sagittal configuration

During inverse kinematic these two parameters are needed to be calculated, which are horizontal and vertical positions of the hip. Similarly foot includes parameters such as horizontal displacement of ankle (x_a), vertical displacement of ankle (z_a), time of total traveling which is single and double support phase (T_c), time of double support phase (T_d), the time that ankle joint reaches maximum height (T_m), step numbers (k), Maximum height on ankle joint (H_{ao}), the horizontal distance between start point and ankle joint for maximum height of ankle joint (L_{ao}), length of half a step Ds, angles of foot lift and contact with the

level ground $(q_b \& q_f)$; and the angles of ground initial terrain $(q_{gs} \& q_{gf})$,these parameters are used to generate ankle and foot trajectory.

Some of these parameters are also illustrated in *Fig. 3.12*. Additionally, there are two extra important parameters which are named $x_{sd}$and x_{ed}. The parameter x_{sd} is the distance between the ankle joint and hip of the support leg, horizontally, in the beginning of the double supports; and the parameter x_{ed} is the distance between the hip and ankle joints, horizontally, at the end of the double support phase. These parameters are shown in Fig. 3.13.

$$t = kT_c \qquad t = kT_c + T_d \qquad t = (k+1)T_c$$

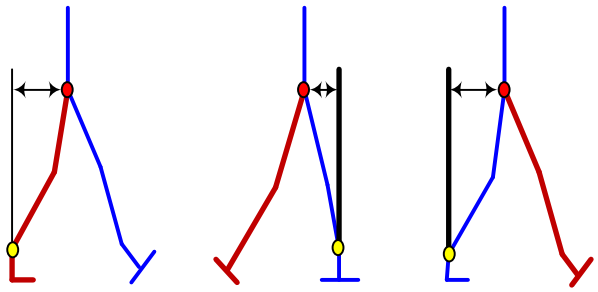


Figure 3.13: Parameters of hip (x_{ed}, x_{sd})

Foot configuration is the other relevant parameters which plays an important role in walking. In this case, foot parameters are shown in Fig. 3.14. This configuration is similar to the human's foot. During a real walking process foot and ankle should be concerned. Therefore, in an ankle trajectory, this configuration is also important.

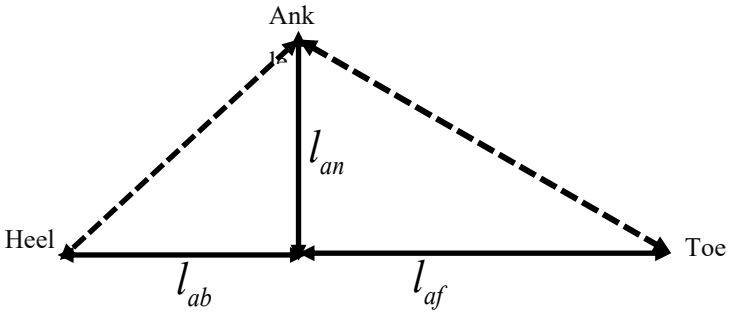


Figure 3.14: Foot configuration

In Fig. 3.14, l_{ab} is the distance between the ankle and the heel, $l_{af}$is the distance between the ankle and the toe; and $l_{an}$is the height of the ankle.

Ankle trajectory; When the foot is traveling, considering the geometrical constraints and avoiding collisions, it is important to consider equation (3.53) and (3.54) conditions for angular position and velocity of the ankle joints.

$$\theta_a(t)=\begin{cases}-q_{gs} & t=kT_c\\ -q_b & t=kT_c+T_d\\ q_f & t=(k+1)T_c\\ q_{gf} & t=(k+1)T_c+T_d\end{cases} \quad (3.53)\qquad \dot{\theta}_a(t)=\begin{cases}0 & t=kT_c\\ 0 & t=kT_c+T_d\end{cases} \quad (3.54)$$

Horizontal and vertical displacement of swing legs during walking is calculated from (3.55) and (3.56).

$$x_a(t)=\begin{cases}kD_s & t=kT_c\\ kD_s+l_{an}\sin q_b+\ldots & t=kT_c+T_d\\ l_{af}\left(1-\cos q_b\right) & \\ kD_s+L_{ao} & t=kT_c+T_m\\ (k+2)D_s-l_{an}\sin q_f-\cdots & t=(k+1)T_c\\ l_{ab}\left(1-\cos q_f\right) & \\ (k+2)D_s & t=(k+1)T_c+T_d\end{cases} \quad (3.55)$$

$$z_a(t)=\begin{cases}h_{gs}+l_{an} & t=kT_c\\ h_{gs}+l_{af}\sin q_b+l_{an}\cos q_b & t=kT_c+T_d\\ H_{ao} & t=(k+1)T_c\\ h_{ge}+l_{ab}\sin q_f+l_{an}\cos q_f & t=kT_c+T_m\\ h_{ge}+l_{an} & t=(k+1)T_c+T_d\end{cases} \quad (3.56)$$

where l_{ao} is the horizontal distance traveled between the ankle joint and the start point when the ankle joint has its maximum height.$h_{gs}$and h_{ge} are the height of ground initial terrain. Both of these two parameters are zero because of the lack of ground's inclination gamma. z_a is the vertical position of the ankle. The horizontal and the vertical velocity of the ankle in landing are zero and therefore, the horizontal and the vertical velocity of the ankle in landing are zero and therefore,

$$\dot{x}_a(t)=\begin{cases}0 & t=kT_c\\ 0 & t=(k+1)T_c+T_d\end{cases} \quad (3.57)$$

$$\dot{z}_a(t)=\begin{cases}0 & t=kT_c\\ 0 & t=(k+1)T_c+T_d\end{cases} \tag{3.58}$$

Hip Trajectory; The hip trajectory is calculated via (3.59) and (3.60).

$$x_h(t)=\begin{cases}kD_s+x_{ed} & t=kT_{c,}\\ (k+1)D_s-x_{sd} & t=kT_c+T_d\\ (k+1)D_s+x_{ed} & t=(k+1)T_c,\end{cases} \tag{3.59}$$

where H_{hmin} and H_{hmax} are the minimum and the maximum height of the hip, x_{ed} is the horizontal distance between the hip and the ankle's joint at the time of kT and x_{sd} is the horizontal distance between the hip and the ankle joint at the time of kT_c+T_d.

$$z_h(t)=\begin{cases}H_{h\min} & t=kT_{c,}\\ H_{h\max} & t=kT_c+0.5(T_c-T_d),\\ H_{h\min} & t=(k+1)T_c,\end{cases} \tag{3.60}$$

Inverse Kinematics

The kinematic parameters for the seven-links biped robot can be seen in Fig. 3.15. The ankle trajectory and hip trajectory are known and therefore, the related parameters of the inverse kinematics are calculated as (3.61), (3.62), (3.63) and (3.64).

$$l_f=\sqrt{(x_h-x_{af})^2+(z_h-z_{af})^2} \tag{3.61}$$

$$\varphi_1=\cos^{-1}\left(\frac{l_1^2+l_f^2-l_2^2}{2l_1l_f}\right) \tag{3.62}$$

$$\varphi_2=\cos^{-1}\left(\frac{l_2^2+l_f^2-l_1^2}{2l_2l_f}\right) \tag{3.63}$$

$$\gamma=\cos^{-1}\left(\frac{x_h-x_{af}}{l_f}\right) \tag{3.64}$$

where, l_f,φ_1 and γ are shown in Fig. 3.15.

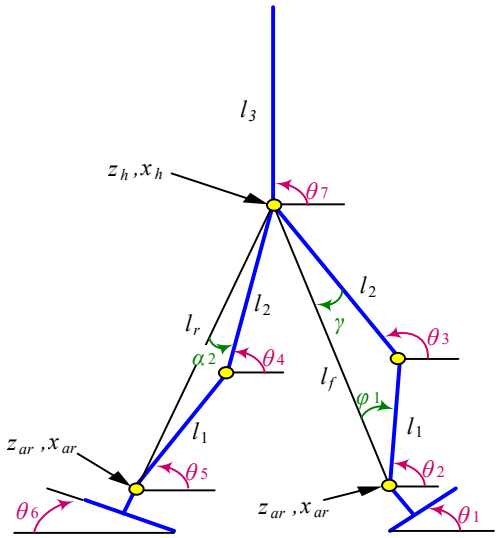


Figure 3.15: Parameters of seven-links biped robot

Equation (3.65) and (3.66) are the left shank and thigh angles

$$\theta_2 = \gamma + \varphi_1 \qquad (3.65) \qquad \theta_3 = \gamma + \varphi_2 \qquad (3.66)$$

Moreover, using the similar manner, the related parameters of the right shank and right thigh angles are obtained by (3.67), (3.68), (3.69) and (3.70).

$$l_r = \sqrt{\left(x_h - x_{ar}\right)^2 + \left(z_h - z_{ar}\right)^2} \qquad (3.67) \quad \alpha_1 = \cos^{-1}\left(\frac{l_1^2 + l_r^2 - l_2^2}{2 l_1 l_r}\right) \qquad (3.68)$$

$$\alpha_2 = \cos^{-1}\left(\frac{l_2^2 + l_r^2 - l_1^2}{2 l_2 l_r}\right) \qquad (3.69) \quad \beta = \cos^{-1}\left(\frac{x_h - x_{ar}}{l_r}\right) \qquad (3.70)$$

By using (3.67), (3.69) and (3.70), the right shank and thigh angles are obtained as

$$\theta_5 = \beta - \alpha_1 \qquad (3.71) \qquad \theta_4 = \beta + \alpha_2 \qquad (3.72)$$

Kinematic Modeling for the Biped Robot with Arms and Forearms

The kinematic parameters for the eleven-links biped robot are shown in Fig. 3.16. This figure represents the improvement of Fig. 3.15 by adding arms and forearms.

Therefore, all position vectors and related angles can be obtained from Fig. 3.16 thus the position vectors for the sake of brevity are given in the Appendix A. They were used for the reason of that these are good functions for specifying trajectory of each joint, since splines are differentiable and easy for evaluating.

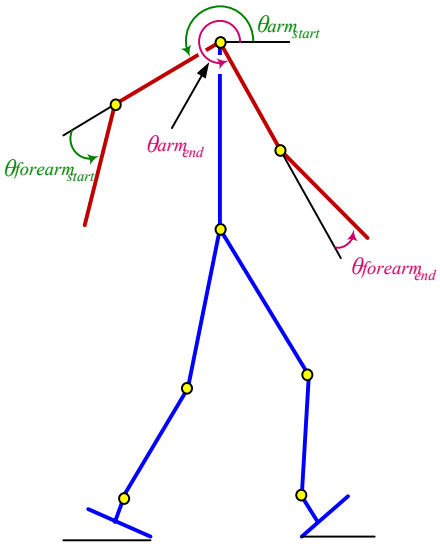


Figure 3.16: Parameters of the improved eleven-links biped robot

3.2.2 Model Predictive Control

The different types of control strategies can be utilized in order to control the legs of the biped humanoid robots, as the humans the brain calculates a trajectory for the legs and sends signals to the muscles in order to contract the muscles, thereby making the leg follow the certain kinds of trajectory and resulting in dynamic walk likewise for the robots, which are controlled by a series of motors acting as the robot muscles, a reference is required. This is usually either velocity, position or torque and a controller to make the motors follow the reference setpoints. The commonly used controllers are PID, LQR and MPC. Each controller has its own set of advantages and disadvantages and depending on the what type of system the most suitable controller is chosen.

A highly nonlinear dynamical system of a humanoid walking robot that relies strongly on contact forces between its feet and the ground in order to realize stable motions, but these contact forces are unfortunately severely limited. Classical trajectory tracking control laws are structurally unable to deal with such strong constraints on the dynamics of a system, especially when having to face strong perturbations. It globally amounts to solving online a sequence of optimal control problems. Such a scheme has been already applied successfully to biped walking robots, but apart from the question of computation time which needs to be taken care of carefully (these schemes can be highly computation intensive), the question of which optimal control problems should be considered to be solved online in order to lead to the desired walking movements is still unanswered. This question is of particular interest because of the limited horizon over which computations can be carried out: the problem is to find which optimal control problem can lead to stable long-term motions while being

solved only over short horizons. The humanoid robot controller should have a sufficient controller sampling rate to follow a gait reference. Meaning, it should command the motors at a sufficient enough rate to prevent the robot from falling.

The state of art humanoid robots such as Atlas by boston dynamics use predictive control to manipulate the system, specifically Atlas uses MPC. The use of predictive control has the advantage of finding the optimal inputs for a finite horizon, based upon a system model of the robot. It has the ability of anticipating future events and act accordingly, to make a smoother walk unlike PID which can only control the current instant. As thesis focuses on creating a biped robot capable of competing with the state of art the most obvious controller would be a predictive controller.

The promising method of biped walking control consists of the ZMP and LIPM which combined with preview control. This method of the combination of the ZMP and LIPM approaches aim to remove some of the drawbacks of both methods. Using the ZMP approach alone leads to puts high torque demands on the actuators of the ankle, in order to keep the feet at exact positions. In other side the LIPM does not suffer from the same drawback, however when using the LIPM the position of the feet is not directly controlled.

This makes the LIPM ill-suited for applications where the exact position of the feet is important, such as walking in terrain with obstacles where the feet must be placed at specific points. By using the method that mixes the ZMP and LIPM approach, arbitrary foot positioning becomes possible. The LIPM presented in 3.2.1, although useful, remain an approximation of a dynamic model of the robot during walk. However, the COM position depends on the configuration of the limbs of the robot. If significant enough, the COM position error can cause the robot to fall, as the ZMP calculated by the simple LIPM model may not be an actual ZMP. A method of handling this model inaccuracy is by preview control. In order to do so, knowledge is required of the expected ZMP error between the LIPM and the multibody model, this error is found by comparing the ZMP reference against the real ZMP measured from the dynamical system. The expected future ZMP error is then stored in a memory buffer. This knowledge can then be used by a preview controller to compensate for the errors caused by the model inaccuracies of the LIPM. For the implementation of the preview controller, it should be noted that the necessary amount of compensation and size of preview horizon correlate. A practical example of this is when the robot walks on what is assumed to be a flat ground, but it actually has a slope. This will

cause an error in the estimated ZMP when compared to the measured ZMP which the predictive controller must compensate for.

The first step to apply the preview control method proposed Katija et al, is to rewrite the LIPM equation 5.3 on page 48 such that the ZMP becomes the output. This yields equation 3.73 which, given position of the COM x, y, zc, gravity g and accelerations of the COM $\ddot{x}$, $\ddot{y}$ results in the ZMP.

$$\ddot{y} = \frac{g}{z_c}(y - p_y) \quad \ddot{x} = \frac{g}{z_c}(x - p_x) \tag{3.73}$$

Then isolating for ur and up by dividing with $\frac{g}{p_z}$ and moving p_x and p_z to the opposite side of the equal sign, and finally multiplying both sides with -1:

$$p_x = x - \frac{z_c}{g}\ddot{x} \qquad p_y = y - \frac{z_c}{g}\ddot{y} \tag{3.74}$$

The first, two new variables named $\dddot{x}$, $\dddot{y}$ are created, which are the time derivative of $\ddot{x}, \ddot{y}$ respectively. The third derivative of position is often referred to as jerk and is the rate of change of acceleration. The system dynamics are formulated on state space form, as shown in equation 3.75. Here p_x,p_y is the x and y coordinate of the ZMP. Zc is the height of the COM of the robot and g is gravity acceleration. The system dynamics are based on equation 3.74 and shown in equation 3.75.

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \qquad p_x = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\dddot{x}, \frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\dddot{y}$$

$$p_x = \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}, p_y = \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} \tag{3.75}$$

A walking pattern generator can be made using the ZMP system dynamics in equations 3.74 Essentially the walking pattern generator will generate trajectories for the COM of the robot, such that the ZMP references are realized.

3.2.2.1 Linear Quadratic Regulator (LQR)

LQR is an optimal control strategy that seeks to minimize the cost of operating a dynamical system. LQR has been successfully implemented to generate trajectories for the COM of a biped robot. LQR makes use of a quadratic cost function as the one seen in equation 3.76. Here x(k) is the system state, Q1, Q2 are weighting matrices, and $u(k)$ is the input signal. The LQR problem is solved either with an infinite or finite horizon. For the infinite horizon LQR, it can be said that the solution is globally optimal.

$$L = \sum_{k=0}^{\infty} x(k)^T Q_1 x(k) + u(k) Q_2 u(k) \tag{3.76}$$

The optimal controller is described using equation 6.5 where the gain L(k) is found by solving the discrete time of the algebraic Ricatti equations.

$$u(k) = L(k)x(k) \tag{3.77}$$

However, a basic drawback of the LQR control strategy is that although an optimal controller can be found, the gain L(k) is just calculated prior to execution, where it is run once and then calculates the desired control for the plant for all future iterations. This means that should model errors exist, or external disturbances affect the system, the gain will no longer result in optimal control. Since it only runs once there is no error correction on the plant thus robustness is low and it leads to the required control output in to the questions.

3.2.2.2 Model Predictive Control MPC

Unlike the LQR, MPC just recalculates the gains for each new sample of execution. The downside to this, is that it significantly increases the computational requirements to execute the algorithm. For the embedded systems, where computational power is often scarce, MPC should be considered with caution. So as an MPC consider the computational requirements increase with the number of system states, as well as by the size of the horizon to be controlled.

The MPC uses a fixed horizon, the required length of which is determined by system dynamics and expected disturbances. Since the MPC algorithm is executed for each new time step, the horizon is said to be receding. Since the horizon is finite, it can only be guaranteed that the MPC provides a locally-optimal controller, in the sense that it is only guaranteed optimal for the given time horizon. As is also the case for LQR, MPC is

optimized with respect to a quadratic cost function, such as in equation 3.78. Here the tune-able parameters are $Q(i)$ and $R(i)$ which are weighting matrices that specify the cost of particular states and inputs. Additional tune-able parameters are H_p, H_u and H_w which are the prediction horizon, control horizon and window parameter respectively. Notice that the mentioned parameters do not depend on time. $\hat{z}(k+i|k)$is the measured system output, $r(k+i\,|\,k)$ is the reference signal and $\Delta\hat{u}(k+i|k)$is the input? The Δ of the input implies the use of integral action to combat steady state error.

$$V(k) = \sum_{i=H_w}^{H_p} \| \hat{z}(k+i\,|\,k) - r(k+i\,|\,k) \|^2_{Q(i)} + \sum_{i=0}^{H_u-1} \| \Delta\hat{u}(k+i\,|\,k) \|^2_{R(i)} \qquad (3.78)$$

When choosing the prediction horizon Hp, the size of it should be larger than the slowest dynamics of the system. Looking past the start-up phase, and walking at a constant pace, the biped gait repeats the same trajectories over and over. At some point increasing the horizon would no longer affect the optimality of the system and only increase computational power, then the horizon should be shortened. Tests in the Simulink environment would indicate the turning point for performance/computational power use. MPC can also take into account the constraints of the system states and input. The constraints are linear inequalities as seen in equation 3.78. These represent the constraints on the actuator slew rate1, actuator range and constraints on the manipulated variable.

$$E\begin{bmatrix}\Delta u(k)\\ 1\end{bmatrix} \le 0, F\begin{bmatrix}u(k)\\ 1\end{bmatrix} \le 0, G\begin{bmatrix}z(k)\\ 1\end{bmatrix} \le 0 \qquad (3.78)$$

System Constraints

The maximum joint velocity and acceleration could be inserted as constraints in the controller with weights indicating the amount of softening for each individual constraint. Hard constraints can also be used, for example limiting a joint angle.

Constraint Softening

The robot has multiple physical constraints, such as joint limits and slew rate of the actuators. These are considered hard constraints, as the robot cannot physically violate them. Setting a constraint as a hard constraint, means that the solution must at all times satisfy the constraint. If this is not possible, the solution becomes infeasible. However, constraints such as velocity of the COM is suitable for softening. Say it is desirable for the COM of the robot to move at a specific velocity. Should a situation arise, where a feasible solution that fulfil the velocity demand is not found, then it is unlikely that a catastrophic failure occurs. On the other hand,

should a situation arise where the MPC produces a trajectory that violates a joint limit of the robot, then it is crucial that the solution is deemed infeasible. Soft constraints should be used with caution, as some system dynamics may be hard constrained in practice.

3.3 Designing Path Planning System for Dinkinesh Biped humanoid robot

In order to Dinkinesh have a high mobility of autonomous functionality in structured and unstructured environment, it must have a navigation system this one includes the path finding and planning in order to deliberate the tasks from the starting to the goal point. In this work the path planning of the biped humanoid robot based on the model predictive control based on the zero-moment point.

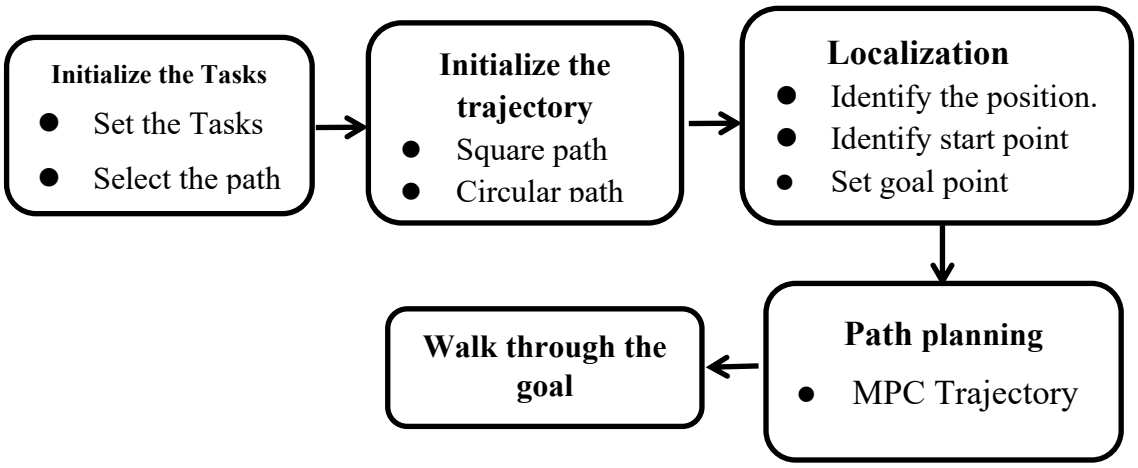


Figure 3.17: Proposed method to design path planning system

First the initialized tasks were selected from the environmental data that perceived from the sensor reading and this data grafted into the other set of modules called the initialized trajectory class, that trajectory class contained of the path of square path, circular path and star path so that the localization of start point and goal points was play the other important subjects for the path planning based on the MPC trajectory then finally the walking to the goal points, such that it contained of the well-known path of the robot to walk through the goal of the biped humanoid robot to reach based on the time frame of time schedules.

3.4 Designing artificial intelligence method for biped walking robot

The first approach in designing the artificial intelligence system in biped walking humanoid robot is to design the reinforcement learning in which a robot trained with a neural network with learn to walk based on the given policy of RL algorithm. The proposed method to deploy AI methods in Dinkinesh humanoid robots is like the deployment of artificial neural networks. To integrate the robotics and artificial intelligence together as shown above in the

figure 3.18 the real world environment extracted from the sensor devices like camera, sonar, stereo and this sensor data can be represented by mapping into 2D-dimensional or 3D-dimensional grids and then after artificial intelligence or machine learning method manipulated for the prediction, inference, data analysis and interpretation to reason for action, execution, navigation and goal reasoning and recirculation frequency occurred in repeating manners.

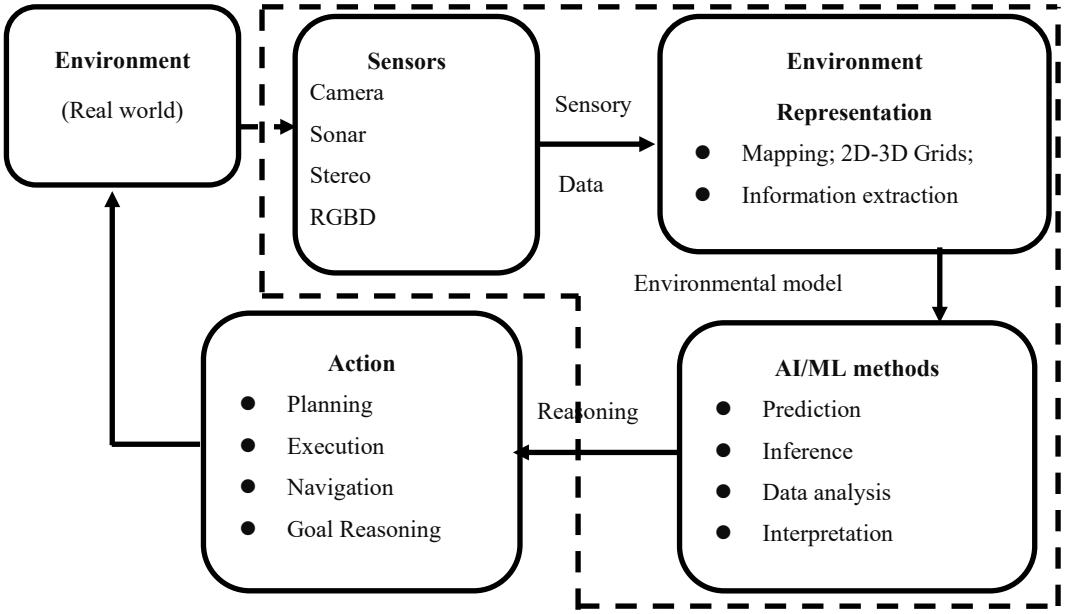


Figure 3.18: Proposed method to Integrate Robotics and Artificial Intelligence

3.4.1 Deep Reinforcement Learning

Human rely on learning from an interaction repeated trial and errors with small variation and finding out what works and what's not. Let's consider an example of child learning to walk that's it tries out various possible movements.

So as to going in through this ML type first of all lets to start by introducing, what it means by itself, so a reinforcement learning is a type of machine learning method that trains an 'Agent' through a repeated interaction with an environment. It works through trial-and-error process that use a reward system to maximize the success. In this thesis the fundamental objectives are to train or teach a walking biped humanoid robot to follow a straight line using a certain reward function using reinforcement learning. From my objectives let's to solve this problem in the traditional ways so as to compare with a new approach of machine learning approach which described in the section after this.

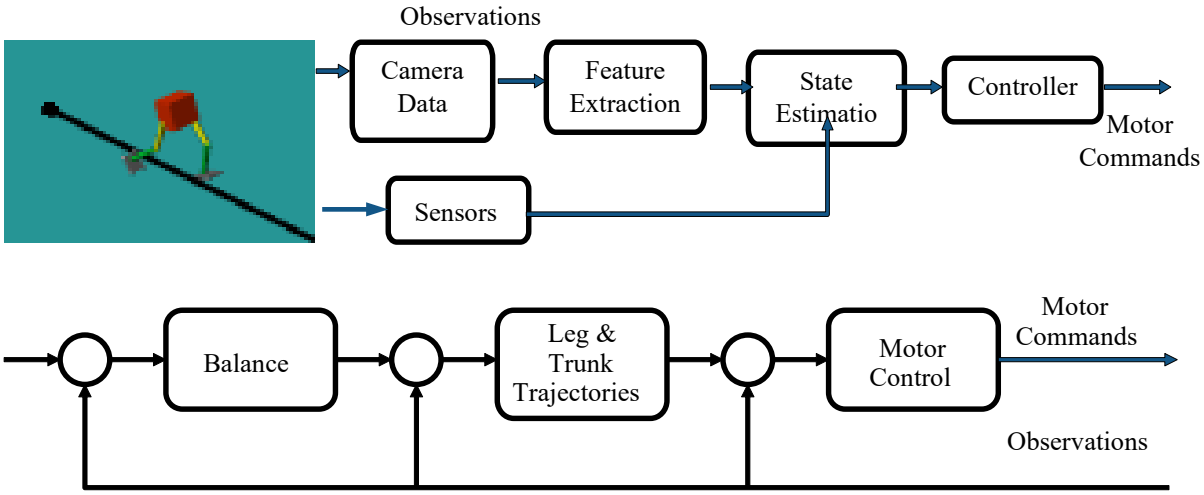


Figure 3.19: Biped walking robot control with traditional approach

The alternative approach to control the biped walking robot can be represented by the following block diagram below in the Figure 3.20, So, how a reinforcement learning approach should consider the biped walking humanoid robot to follow a certain kind of path of walking in order to replace the traditional control method. To understand how the RL should be mapped for the given state action to be performed without knowing the environment dynamics this can be represented by the *figure 3.21* below.

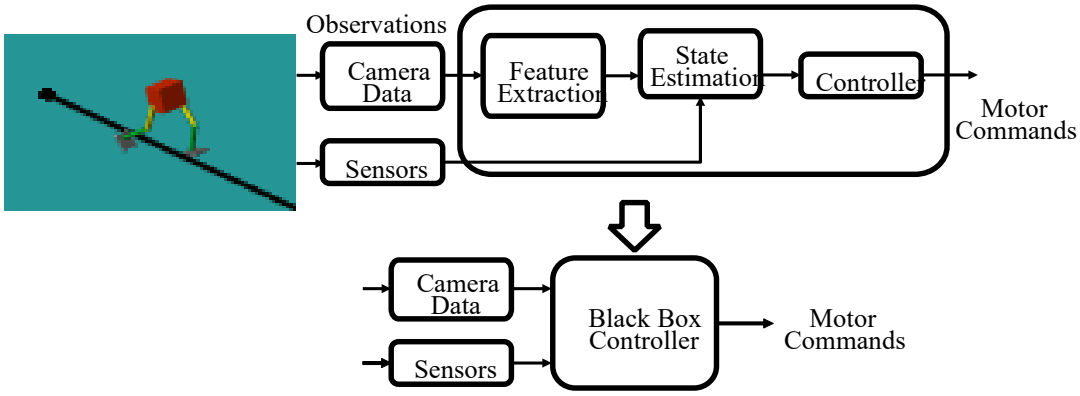


Figure 3.20: Biped walking robot control with alternative approach

Every reinforcement learning problem can be represented by the agent and environment interaction in the iterative update of state information, reward action and performance of the environment's variables. This algorithm can be implemented practically in the following diagram below in the figure as shown.

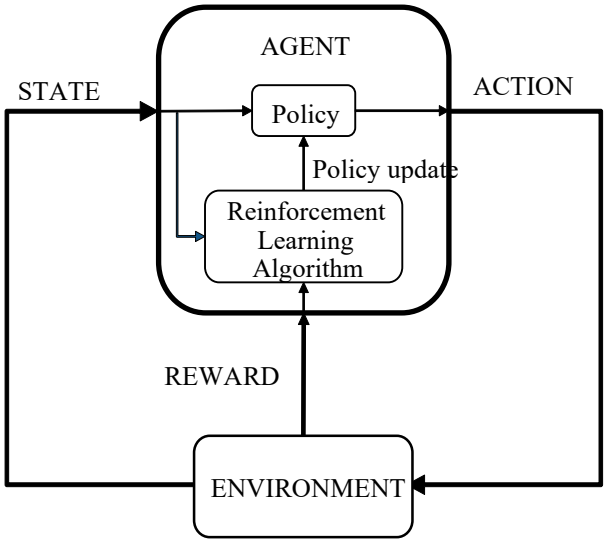


Figure 3.21: A Practical Example of Reinforcement Learning training in biped walking humanoid robot.

The biped walking robot that just acts like computer learns and tells how to a robot should walk as like an agent using a sensor reading from lidar, camera etc. as state in order to represent the path conditions, the biped position as like an environment, by generating walking pattern, stopping in the obstacle, sitting, standing command in the form of action in which based on an internal state to the action mapping (policy) that tries optimize a walking stability and more like human like walking by allowing efficient energy consumption like a reward. The policy is updated through a repeated trial and error by reinforcement learning algorithm.

3.4.1.1 Deep Deterministic Policy Gradient (DDPG)

DDPG is an actor-critic algorithm based on deterministic policy gradient. The DPG algorithm consists of a parameterized actor function $\mu(s|\theta\mu)$ which specifies the policy at the current time by deterministically mapping states to a specific action. The critic $Q(s, a)$ is learned using the Bellman equation the same way as in Q-learning. The actor is updated by applying the chain rule to the expected return from the start distribution *J* with respect to the actor parameters. I had considered a standard reinforcement learning setup consisting of an agent interacting with an environment E in discrete timesteps from the continuous state action pairs. At each timestep t the agent receives an observation x_t, takes an action a_t and receives a scalar reward r_t. In all the environments considered here the actions are real-valued $a_t \in \mathbb{R}^N$. In general, the environment may be partially observed so that the entire history of the observation, action pairs $s_t = (x_1, a_1, \ldots, a_{t-1}, x_t)$ may be required to describe the state. Here, I assumed the environment is fully-observed so $s_t = x_t$.

An agent's behavior is defined by a policy, π, which maps states to a probability distribution over the actions $\pi: s \rightarrow p(A)$ the environment, E, may also be stochastic or deterministic. It can be possible to model it as a Markov decision process with a state space S, action space $A = \mathbb{R}^N$, an initial state distribution $p(s_1)$, transition dynamics $p(s_{t+1}|s_t, a_t)$, and reward function $r(s_t, a_t)$. The return from a state can be defined as the sum of discounted future reward $R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i)$ with a discounting factor $\gamma \in [0,1]2\ [0; 1]$. Note that the return depends on the actions chosen, and therefore on the policy π, and may be stochastic. The goal in reinforcement learning is to learn a policy which maximizes the expected return from the start distribution $J = E_{r_1, s_t \sim E, a_t \sim \pi[R_1]}$. I denote the discounted state visitation distribution for a policy π as ρ^{π}. The action-value function is used in many reinforcement learning algorithms. It describes the expected return after taking an action at in state st and thereafter following policy π:

$$Q^{\pi}(s_t, a_t) = \mathrm{E}_{r_{i\geq t}, s_{i\geq t} \sim E, a_i > t \sim \pi}[R_t | s_t, a_t] \tag{3.79}$$

Many approaches in reinforcement learning make use of the recursive relationship known as the Bellman equation:

$$Q^{\pi}(s_t, a_t) = \mathrm{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma \mathrm{E}_{a_{t+1} \sim \pi}\left[Q^{\pi}(s_{t+1}, a_{t+1}) \right] \right] \tag{3.80}$$

If the target policy is deterministic, we can describe it as a function $\mu: S \leftarrow A$ and avoid the inner expectation:

$$Q^{\mu}(s_t, a_t) = \mathrm{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, \mu(a_{t+1})) \right] \tag{3.81}$$

The expectation depends only on the environment. This means that it is possible to learn Q^{μ} off policy, using transitions which are generated from a different stochastic behavior policy β. Q-learning is a commonly used off-policy algorithm, uses the greedy policy $\mu(s) = argmax_a\, Q\,(s, a)$. We consider function approximators parameterized by θ^Q, which we optimize by minimizing the loss:

$$L(\theta^Q) = E_{s_t \sim \rho^{\beta}, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t|\theta^Q) - y_t)^2] \tag{3.82}$$

Were

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1})|\theta^Q) \tag{3.83}$$

While y_t is also dependent on θ^Q, this is typically ignored.

The use of large, non-linear function approximators for learning value or action-value functions has often been avoided in the past since theoretical performance guarantees are impossible, and practically learning tends to be unstable. Recently, adapted the Q-learning algorithm in order to make effective use of large neural networks as function approximators.

ALGORITHMS

It is not possible to straightforwardly apply Q-learning to continuous action spaces, because in continuous spaces finding the greedy policy requires an optimization of at a_t every timestep; this optimization is too slow to be practical with large, unconstrained function approximators and nontrivial action spaces. Instead, here we used an actor-critic approach based on the DDPG algorithm. The DDPG algorithm maintains a parameterized actor function $\mu(s|\theta^{\mu})$ which specifies the current policy by deterministically mapping states to a specific action. The critic $Q(s,a)$ is learned using the Bellman equation as in Q-learning. The actor is updated by following the applying the chain rule to the expected return from the start distribution J with respect to the actor parameters:

$$\Delta_{\theta_\mu} J \approx E_{s_t \sim \rho^{\beta}} \left[\Delta_{\theta_\mu} Q(s,a|\theta^{Q})|_{s=s_t, a=\mu(s_t|\theta^{\mu})} \right]$$

$$= E_{s_t \sim \rho^{\beta}} \left[\Delta_a Q(s,a|\theta^{Q})|_{s=s_t, a=\mu(s_t)} \Delta_{\theta_\mu} \mu(s|\theta^{\mu})|_{s=s_t} \right] \quad (3.84)$$

3.5 Hardware System Design and integration

To design the hardware system of Dinkinesh biped humanoid robot, I used the hardware components with Linux operating system of Rasipian buster OS with raspberry pi 4B with 2GB RAM and Arduino mega2560 for the real time motion control with serial communication. In this section the overall hardware system design and integration were discussed accordingly in relevant tools and techniques. In the figure 3.22 as shown in the above the electrical block diagram of the thesis to be implemented for the hardware design and implementation with the state of art of embodying the hardware layers with the abstracts of software program embodiment together were drafted. In this regard all of the hardware components descriptions explained in briefly according to the relevant design specification and constraints.

3.5.1 Electrical System Block Diagram

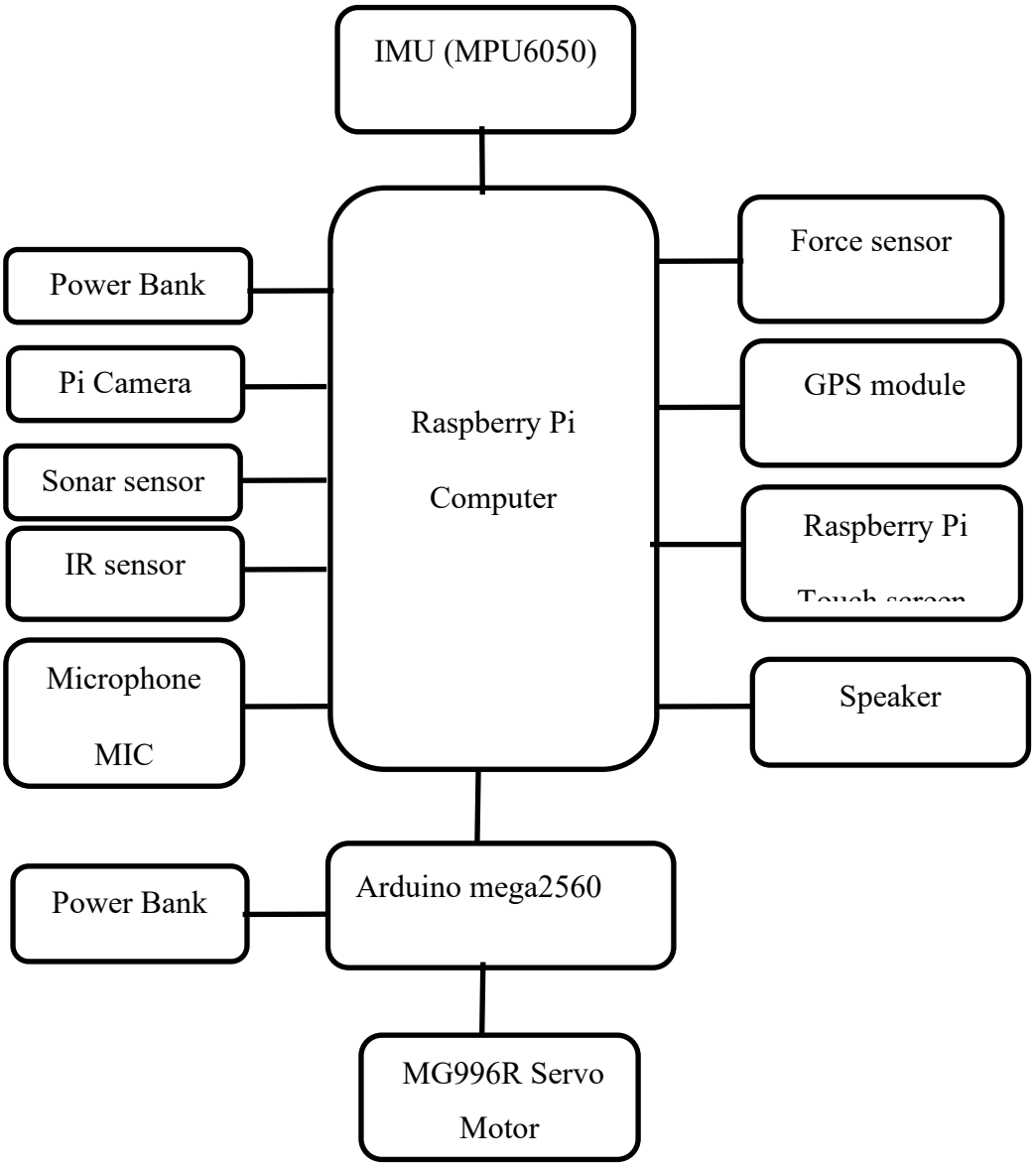


Figure 3.22: Electrical System Block Diagram

3.5.2 Overall Hardware Flowchart Diagram

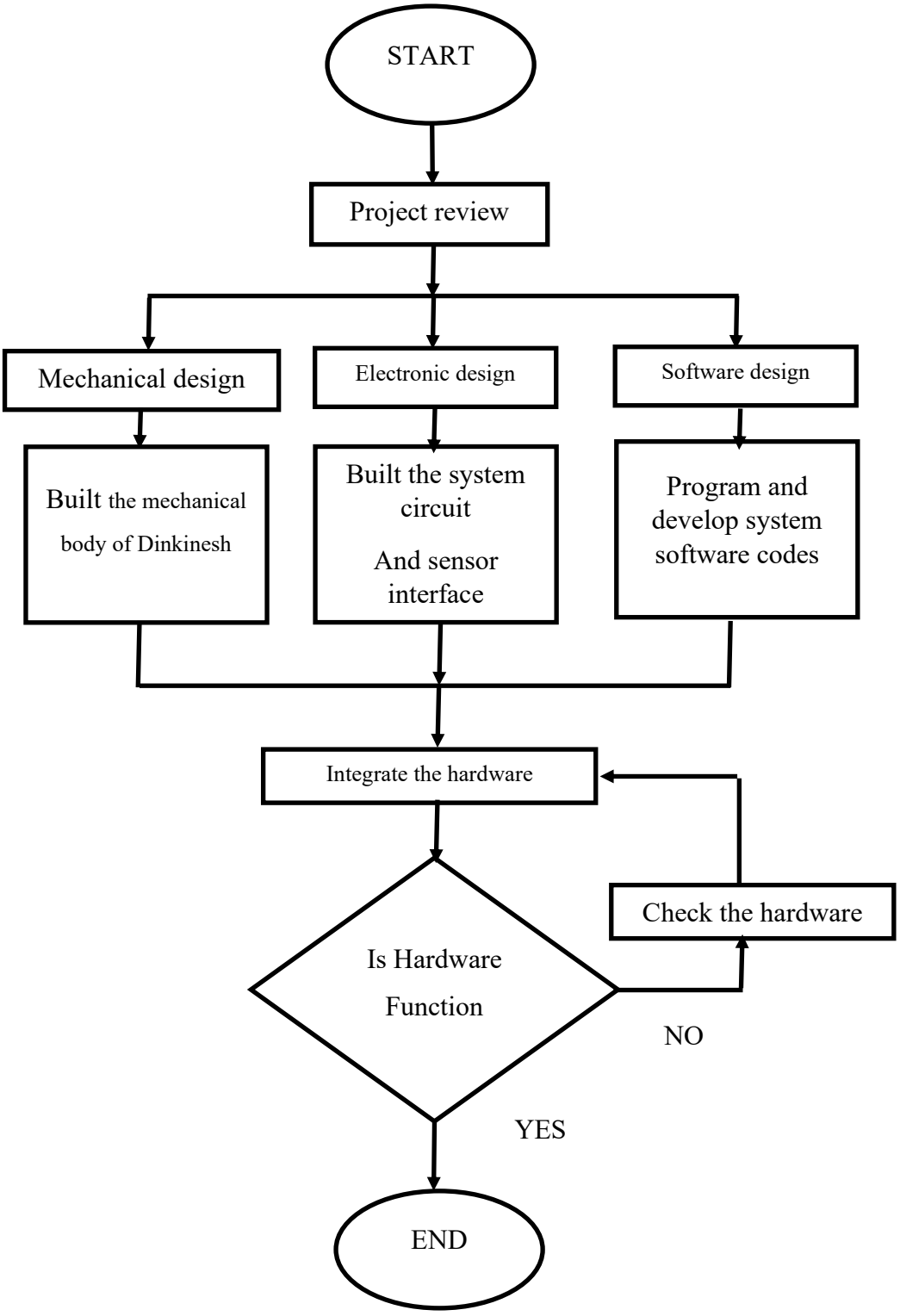


Figure 3.23: Overall Hardware flowchart Diagram

CHAPTER FOUR

SIMULATION RESULTS AND DISCUSSION

4.1 Biped Locomotion and Balancing Strategies of Dinkinesh Humanoid

In this section, I had discussed the fundamental of biped locomotion strategies for the Dinkinesh humanoid robot, that contained of the modules of walking and balancing strategies in the smooth even terrain in the MATLAB Simulink environments, to model the walking system for the humanoid robot Dinqe, I had used two dimensional and three dimensional linear inverted pendulum model which was a physical system of walking robot which allow the stable walking mechanisms and the walking trajectory was designed based on finite time horizon, linear quadratic regulator or just model predictive control problem based on the most amazing biped walking control issue in balancing control called zero moment point(ZMP) and the center of mass trajectory projection.

4.1.1 Designing Linear Inverted Pendulum Model

When a biped robot is supporting its body on one leg, its dominant dynamics can be represented by a single inverted pendulum which connects the supporting foot and the center of mass of the whole robot. In this regard the simulink model of six degree of freedom of biped walking robot, the simulation of the walking biped robot contain the closed form of 6DOF inverse kinematics solution with the input of both leg foot trajectory in addition with the robot joint controller which was modelled with motion dynamics of the biped walking system with the plant model of walking robot, the plant dynamics was designed based on the physical system in MATLAB Simulink environment using sim-scape and sim-physics MATLAB libraries in order to obtain the results in the form of animated simulation.

The first simulation was performed based on open loop two-dimensional linear inverted pendulum and the second simulation as well modelled with closed loop control based on model predictive control of the three-dimensional linear inverted pendulum, both of the simulation model discussed in the section below independently.

The two-dimensional double linear inverted pendulum model to represent the walking constraint of biped humanoid robot in the MATLAB Simulink environment was given in the figure below in which a 12DOF biped walking humanoid robot based on the step trajectory

of the COM and ZMP with the open loop control of the input reference of the center of mass of the robot position of the actuators.

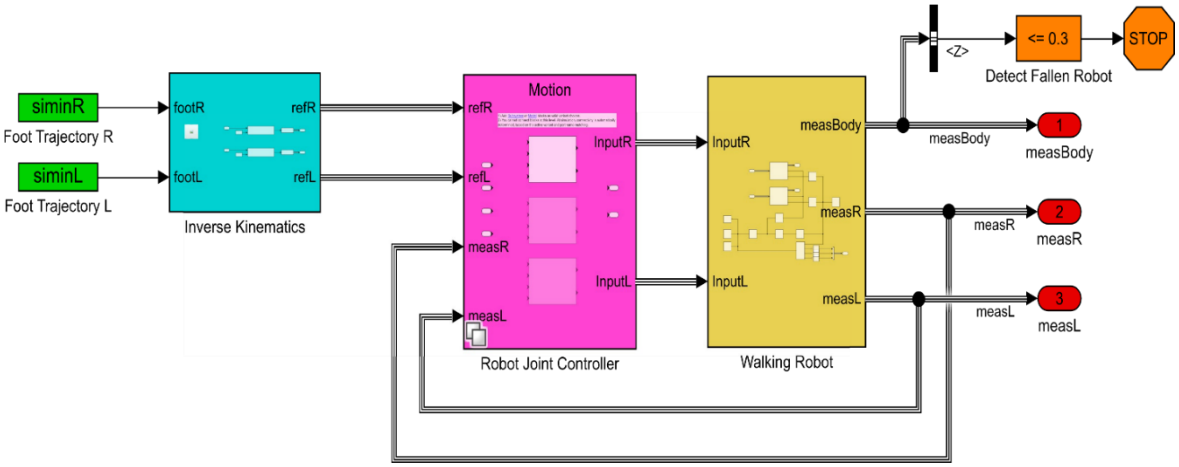


Figure 4.1: 2D double linear inverted pendulum walking robot simulation

The fundamental design philosophy of the above biped walking robot Simulink model was based on the 2D-LIPM which used to simulate the biped walker based on the foot trajectory of the walking robot, for this task first the swing foot trajectory must be generated, in this case smooth optimized interpolation called cubic spline polynomial trajectory technique which was used to design for the both legs symmetrically in MATLAB script function for the swinging foot trajectory, the third order spline provides the position, velocity and acceleration from the given input foot position of the two legs, swinging height, initial step time, final step time and the walking duration of the robot, those variables finally allows the smooth position, velocity and acceleration in the direction of trajectory for x and y, z and for both xy and z direction finally those reference foot trajectory subjected to the closed form inverse kinematics problem which transform the inverse kinematics joint solution from the body frame to the foot frame of the walking robot which transform matrix describing the foot position and orientation with respect to the body position and orientation in which the inverse kinematics problem was obtained by an analytic solution to found a closed-form inverse kinematics joint solution by setting the base of each leg as the end-effector and foot as the base.

So, these ways that allowed to solve the joint angles in reverse order. This way the three joints (hip pitch, roll, yaw) intersect at a point. Therefore, in this case, I had performed offset to transform body to foot for the base of each leg to foot. Then I found the inverse of the transform to get foot to base of each leg and finally the analytic solution of each joint angles obtained. After finding the analytical solution of each joint angles for the two legs, I had

animated a symmetrical walking pattern model by simulating a linear inverted pendulum model (LIPM) with a simple foot changing controller to create body trajectory based on the LIPM by assuming two conditions. First the walking pattern was symmetrical and the second also there was no input disturbances in the plant. In the context of simulating a linear inverted pendulum with a simple foot changing controller by allowing the model parameters with the height of the robot when standing straight was defined as z-height of the robot. The height of the pendulum (and hence the height at which made the robot walk at) was defined by Z-height of the model. Then the swinging height of the foot trajectory and finally then it can be possible design a discrete state-space model of the linear inverted pendulum.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{g}{zModel} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{zModel} & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.1)$$

After representing the discreate state space model of the LIPM the model parameter swaps in the single function by setting the foot initial conditions and number of steps. This was crucial for the robot to standing still with straight legs by remembering that it should be optimized by a trajectory for the actual robot and by using a simple model to create a trajectory. The trajectory was simply coming from the natural dynamics of the pendulum (and the initial conditions).

So, it can be simple to choose the next foothold and let the pendulum give us the trajectory. The LIPM foot changing controller assumes the robot makes the left step first and the first two entries which define the starting position of both left and right feet, at offsets of -xTorso and +xTorso, respectively. Then, the third entry of foot position will be for the left foot first, the fourth for the right foot, and so on. Change accordingly depending on which foot was gone out first and also which direction it was headed. So, in order to simulate the symmetrical walking in the first phase move the COM above one foot which was used to create a center of mass (COM) and swing foot trajectory for the robot to follow.

So, it was mandatory to keep in mind that the robot will be standing still at the beginning (zRobot), but it has to lower its center of mass (zModel) and start moving. In the second phase make a half step in order to the robot to go to the initial condition as I sated earlier. Also, we should have to think about how the swing foot will move at the same time. The swing foot moves its other leg to the next plant position where it will become the new base for the pendulum. The pendulum base also where the robot has its plant foot at, this also

shows how far the robot will move its COM to before making the stance transition (right to left or left to right). In the third phase making consecutive steps and this was the core functionality of the walking pattern. Once we have the initial conditions that give us a symmetric trajectory, we can use this to create a pattern. Note how the "stitching" were done using the changing leg. Since we know the trajectory is going to be symmetric, it can be placed that the next foot hold at a mirrored distance away. Then after the simulated model to create the next trajectory, then the foot hold was changed again after step time by the single support time. According to the simulation of walking pattern based on the inverse kinematics problem the walking pattern given by the figure below.

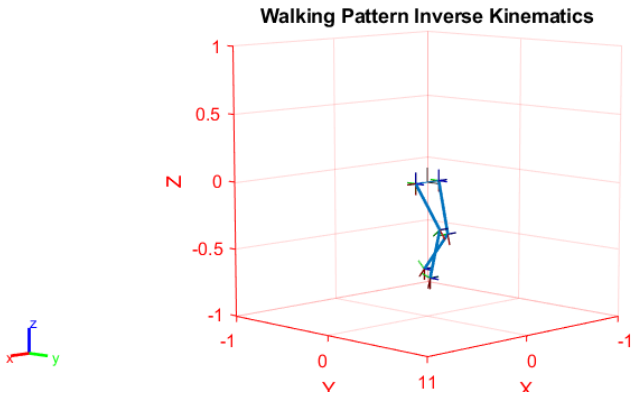


Figure 4.2: Walking Pattern based on inverse kinematics problem with LIPM

So, the objectives weas not to animate the LIPM but also provides the walking simulation that can provide a stable gait pattern generation, for this concern I had just transformed body trajectory into feet trajectory by performing a transformation of reference frames to make the body and swing trajectories, which are defined with respect to the global origin, into feet trajectories with respect to the body (COM).

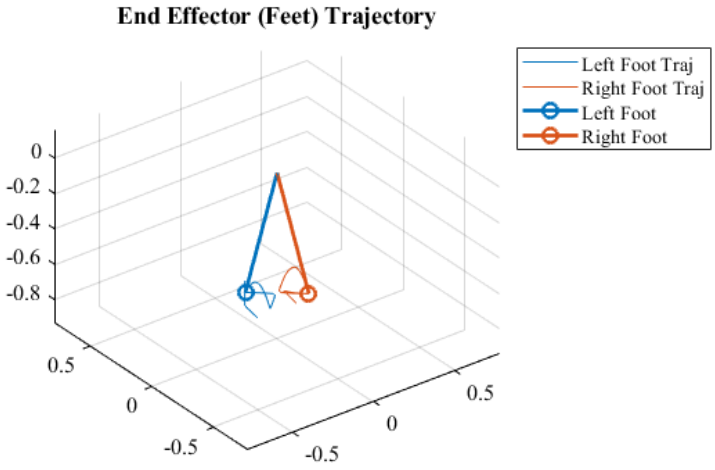


Figure 4.3: The end effector(feet) trajectory of LIPM walking robot

Finally, the two-dimensional linear inverted pendulum model of the biped walker was animated in the sim-mechanics simulation in MATLAB toolbox with help of animation, at the end, the robot responds the flowless walking in the given walking trajectory, this result of the animated walking was given by the figure below.

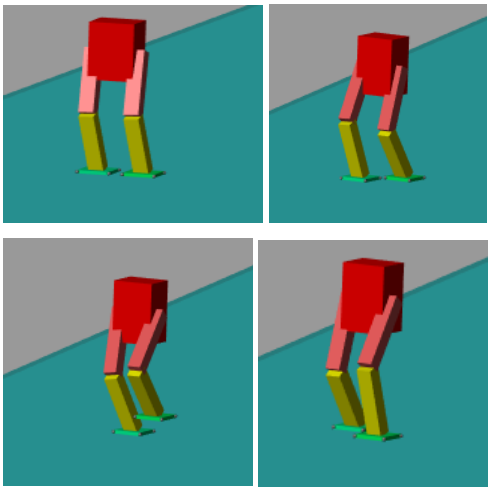


Figure 4.4: 2D LIPM walking pattern of biped robot Animation

The biped locomotion approaches, that I had used in this thesis was designed with the walking system based on the physical system modeling of the 12DOF humanoid legs based on double linear inverted pendulum mode in which it relied on the point mass, that which can allow the robot to walk in center of mass position of the pendulum in relative to the reference point tracking with allowance of ZMP (zero moment point) i.e., this the most advanced concepts in the case of the robots to walks in stable manner with helps of the support polygon of ZMP region. To design the biped locomotion and balance strategies I had used 2D and 3D linear inverted pendulum with a controller of the model predictive control in order to generate walking trajectory for Dinkinesh Humanoid robot. The first steps in modeling the biped walking system were to gone through the calculations of the joint angles using the inverse kinematics and checked with rigid body tree.

Here I had calculated the joint angles from the left and right foot trajectories. For this reason, I had used an analytic solution of inverse kinematics for the leg based on closed-form inverse kinematic joint solution for humanoid robots. The function for inverse kinematics takes the input to the simulation will be the feet transform matrix that contain position and orientation information. Finding the analytic solution to the inverse kinematics requires deriving an expression based on the forward kinematics, which can be time consuming, difficult, and even impossible for some robot geometries and constraints. In that case, we may want to

consider using the optimization-based inverse kinematics solver from robotics system toolbox in the MATLAB. Although it was a slower than the analytical solution, it can be applied to arbitrary kinematic chains with constraints. Based on the inverse kinematics formulation it can be simples to simulate linear inverted pendulum model (LIPM) with simple foot change controller with creating body trajectory based on the LIPM. I had assumed two conditions; the first one was the walking pattern was symmetric and the second was there are no disturbances around the inverted pendulum model due to joints motors. The piece of trajectory used to generate a walking pattern can be found by using MATLAB script which computes to finds initial condition for creating symmetric piece of trajectory.

4.1.2 Designing Model Predictive Control for Dinkinesh humanoid

To design model predictive control for the biped walking trajectory and pattern generation for the humanoid robot, I had used 3D linear inverted pendulum equation from the linearized state space representations, this was because of the walking humanoid robot can be easily represented by 3D-LIPM with most dynamic locomotion's of biped walking humanoid robot with this representation also presented in the figure as shown below in the *figure 4.5.* So, the general design criteria were assumed by taking the point mass pendulum, with regard to driving to the continuous plant model with the discretization of the model into most controllable with receding horizon control also called model predictive control (MPC). Whether one of the legs should be in the support phase or in swinging phase, this was because of more natural walking of human legs follows a procedure of one leg in the swing phase the other legs in support phase, when the biped was walking as like human like walk, this can be achievable by representing with this 3D-LIPM, as shown in the below figure 4.5 in this case the LIPM equation can be represented by the combination of both COM and ZMP output as follows in equation (4.2) and (4.3).

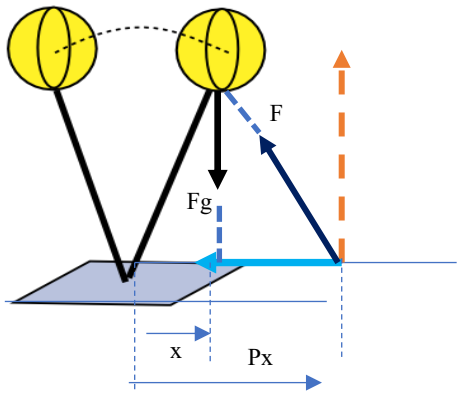


Figure 4.5: The Three-dimensional Linear Inverted Pendulum.

$$\ddot{y} = \frac{g}{z_c}(y - p_y) \tag{4.2}$$

$$\ddot{x} = \frac{g}{z_c}(x - p_x) \tag{4.3}$$

Based on the equations of acceleration along to x and y axis in the equation (4.2) and (4.3) thus can be possible to write the 3D-LIPM equations in the form of the ZMP outputs, as shown in the equation (4.4) and (4.5).

$$p_x = x - \frac{z_c}{g}\ddot{x} \tag{4.4}$$

$$p_y = y - \frac{z_c}{g}\ddot{y} \tag{4.5}$$

From the equations (4.4) and (4.5) I had the three parameters in order to represent the walking humanoid in the form of ZMP outputs, which determine the physical system model interims of gravity constant, the center of mass height which was the model constant and the sampling time constant. To define the model predictive controller in biped walking trajectory generation it was very important to represent the 3D-LIPM equation into state space representation, for the applications of continuous to discreate state space forms. To create state space representation for one axis, First, we need had to obtain a state-space representation where the ZMP was the output. Let's define the input U as the time derivate of the horizontal acceleration of the CoM.

$$\mathrm{U_r} = \frac{\mathrm{d}}{\mathrm{dt}}\ddot{\mathrm{x}} \tag{4.6}$$

From the above equation (4.6) we can introduce the state space representations for the time derivatives of the horizontal acceleration of the CoM of the following dynamical system.

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \tag{4.7}$$

$$p_x = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{4.8}$$

To design the model predictive controller as the optimal control problem for the given dynamical system the plant must be discretize and assigned as a measured/unmeasured output to formulate it, this generation problem as an optimal control problem, let's discretize the continuous dynamical system using MATLAB c2d function in the following equation below:

$$x(k+1) = Ax(k) + Bu(k)$$
$$p(k) = Cx(k)$$

$$A = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} B = \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} C = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \tag{4.9}$$

I should have to specify the measured outputs; this was very important for the MPC control problem formulation. In this case, I had just wanted to track the reference ZMP because, I had used the given walking output in the form of ZMP outputs, this was because of the dynamic stability of the biped walking machine were fundamentally relied on the zero moment point of the walking support area, these ways it can be done for the walking of biped dynamic stability in the locomotion of biped humanoid robot.

In the case of assigning the measured and unmeasured output interims of input/output system. Now it can be possible to design an MPC controller, for the simplicity first MPC for y axis and next for x axis with the following characteristics; like the sampling time, the measured output was ZMP and the unmeasured outputs: the center of mass position and velocity. Prediction horizon and to verify no tracking on second and third output which was also called the center of position and velocity, ZMP reference, preview reference. The MPC as a walking pattern generator for that reason, I had used two parameters for the first term step parameters which contained of step length, step width and step time that was the walking step parameters and the second was the MPC parameters that was used to design the controller for the walking pattern generation which contained of prediction horizon, number of output and initial state of the input variables.

The simple design method that was represented in the walking dynamics of the biped humanoid robot, the walking of humanoid that is based on model predictive control for the trajectory generation and the first design was based the 3DLIPM for the walking trajectory generation which takes the input reference ZMP along the x and y direction and this reference inputs were subjected to the walking controller in order to control the whole walking dynamics of the humanoid robot, the whole walking simulation diagrams as given in the figure below.

In the figure described below the 3D-LIPM constrained MPC based on biped walking pattern generation, the entire Simulink block as represented, we have two blocks interconnected in the closed loop form with two input and two output, those two-block presented in above diagram, one was the controller which filled with light blue color and the other was the plant that was filled with yellow color.

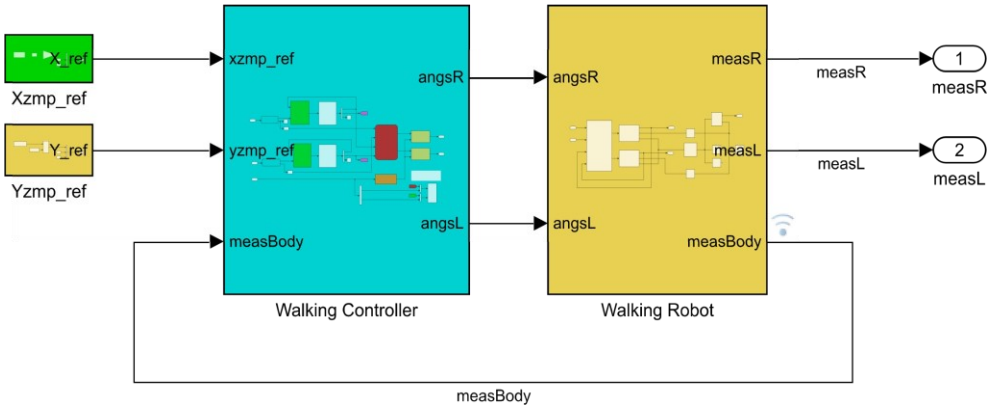


Figure 4.6: The 3D-LIPM MPC biped walking trajectory generation

The walking controller were fundamentally it was very crucial to tell in what pattern the biped robot should walk with accurate reference tracking from the input reference ZMP and get the desired output of right and left legs position and orientation in the complex three-dimensional plane environment accordingly.

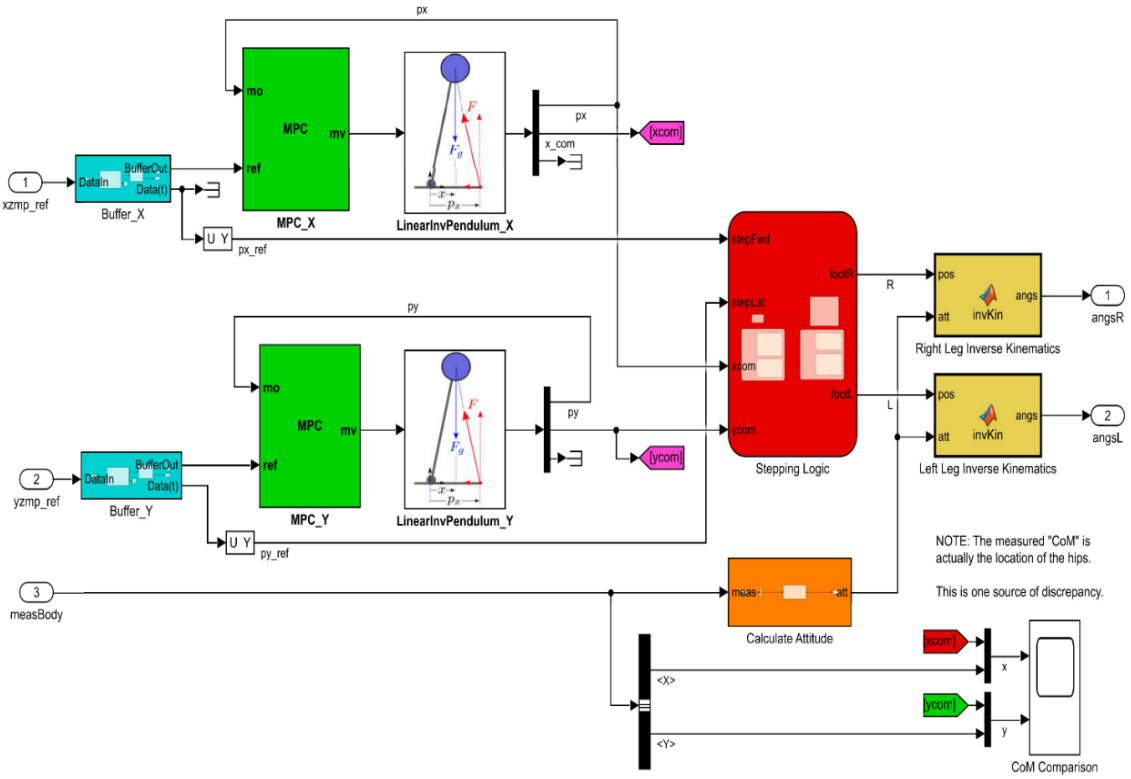


Figure 4.7: 3D MPC based on LIPM biped walker

The good ways to represent this complex dynamical system. as I mentioned before ZMP was an excellent algorithm with the modeled Simulink block in the MATLAB environment which follow the closed loop control of the three-dimensional linear inverted pendulum for the biped walking robot in terms of physical system representation from the given linearized state space model and the model predictive control for the trajectory generation control, from

all the model consideration the MATLAB Simulink were used and sim-mechanics from MATLAB library, to design the block based animated design.

It was very important and mandatory to represent a biped walker robot to follow certain procedure of walking and this walking procedure in MATLAB Simulink can be represented by state flow diagram of step logic diagram, this diagram presented in the figure below which allows the procedure of the biped walking robot transition of legs according to stepping time, length of step and swinging time as well this was depend on the decision of which legs supposed to be active or not.

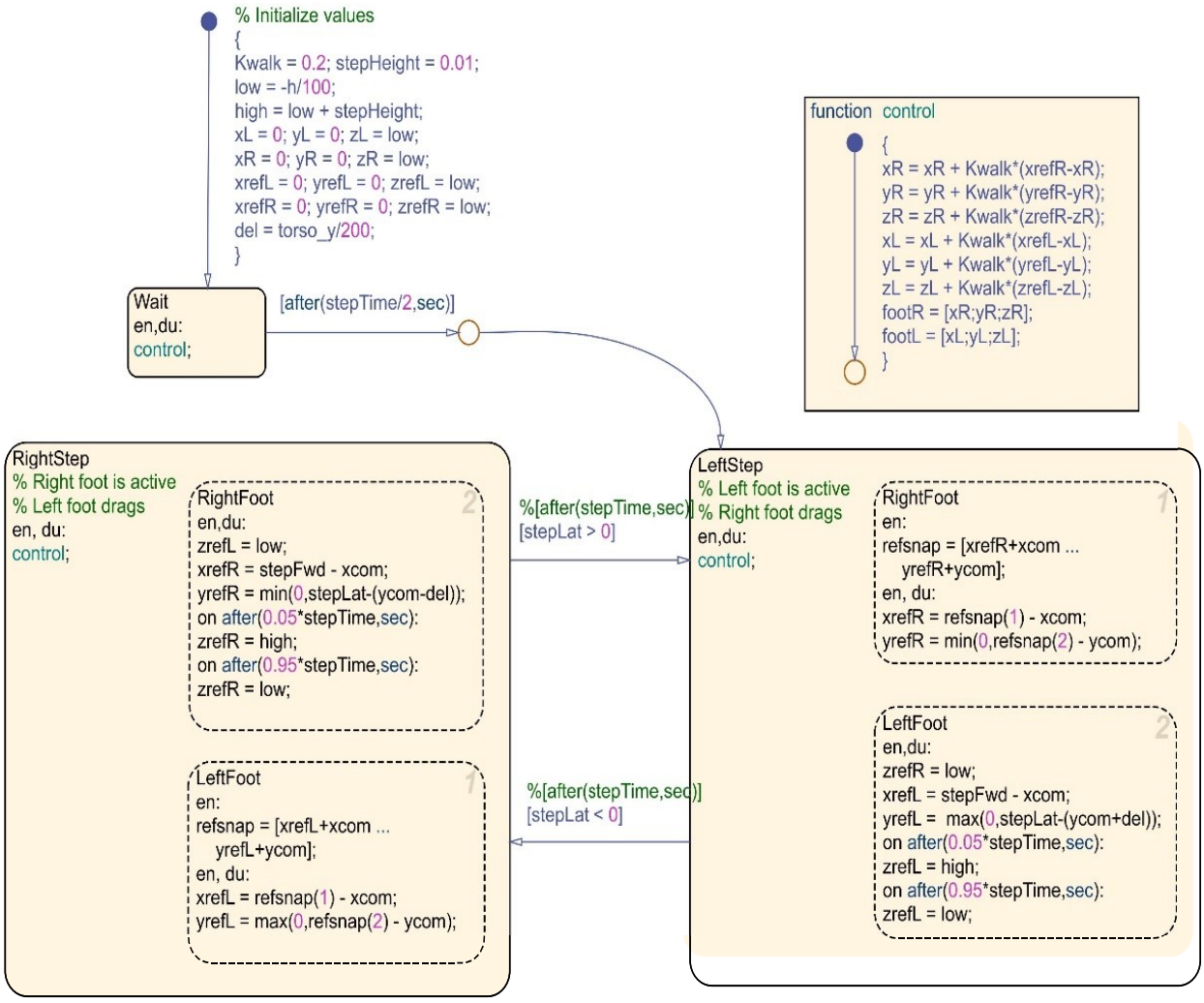


Figure 4.8: Biped walker step logic state diagram

In this case there was two function which allows how two legs should be walk in the active transitions in which it can be represented by the right step and left step by the time boundary of left swing and right swing of the robot legs accordingly. The step logic block provides a useful information to be computed for the inverse kinematics problem for that of both legs of the biped walker. Now here, I would like to show, how a MPC controller can be used in

biped walking robot for the walking trajectory generation of the biped walking humanoid, accordingly in the MATLAB Simulink environment with the application of zero moment point (ZMP) and center of mass (COM) projection to control with an optimal control problem like the model predictive controller as a controller in the trajectory generation, the whole model in this regard given by as shown below in the figure.

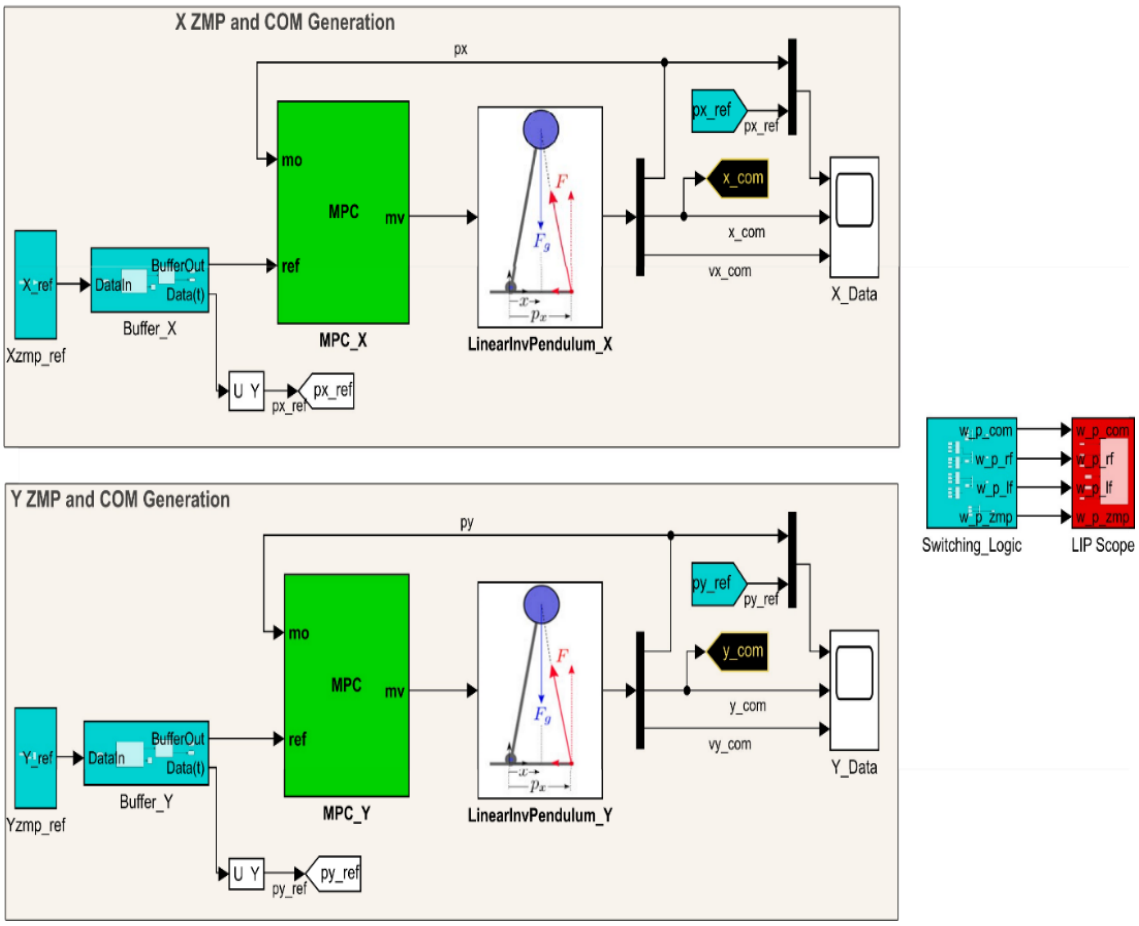


Figure 4.9: Biped walking pattern generation with model predictive control

As shown above in the *figure 4.9*, there are two basic elements in the design of MPC control in the application of trajectory generation, what does I mean two basic element by itself, that was just an utilizing elements to control the BWR, the ZMP and COM generation along to the direction of x and y in which the input reference ZMP buffered into the MPC controller along to the x and y for the walking pattern that subjected to the linear inverted pendulum of the physical walking representation which used to generate the walking pattern with the model predictive control.

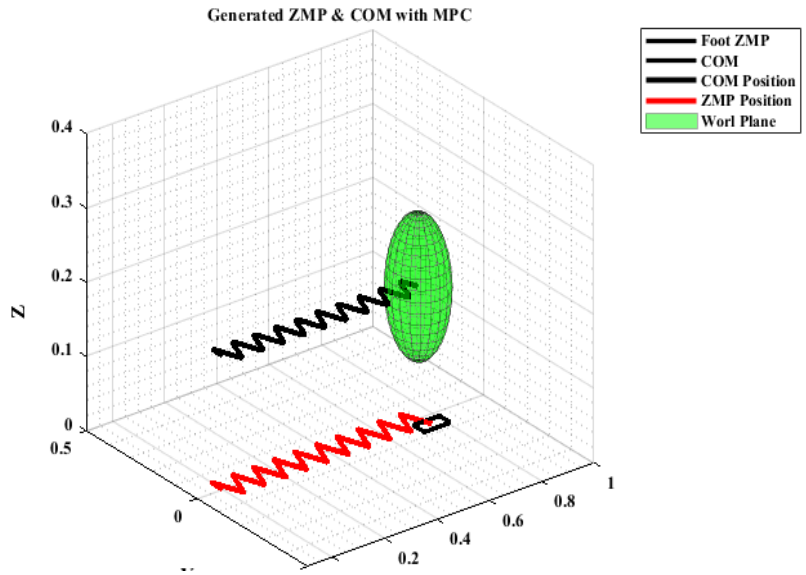


Figure 4.10: BWR generated ZMP and COM with model predictive controller

To see that how an MPC controller generate the trajectory for biped walking robot in the three-dimensional space, I had showed in the figure 4.10 above accordingly specified. So, as a figure shown above *figure 4.10*, we have a center of mass (COM) trajectory and the zero-moment point (ZMP) trajectory along in the x-direction, so as to realize the impact of the x and y data point you can see the figure below for the clear understanding.

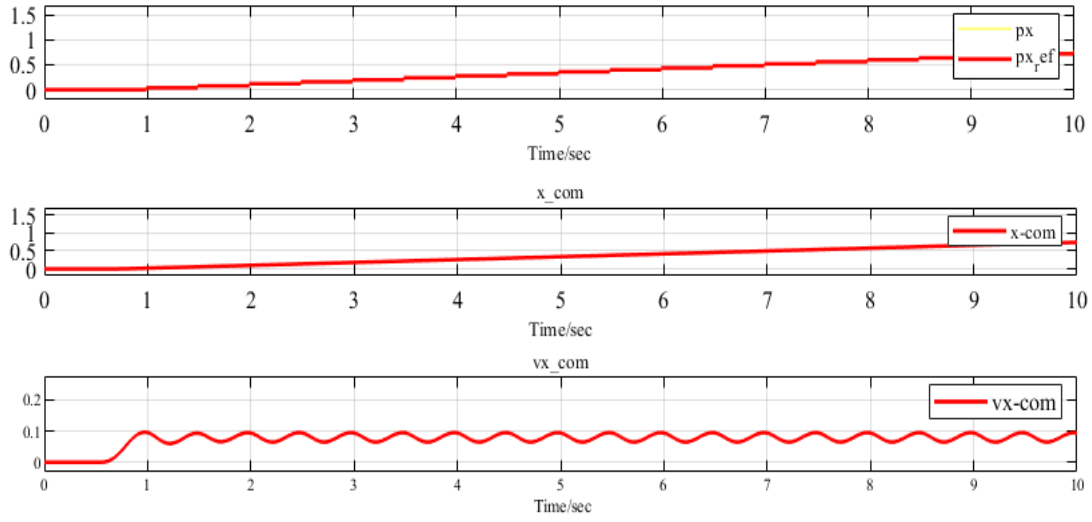


Figure 4.11: MPC generated biped walker right leg data

So, in the above *figure 4.11* we have the parameters in the right leg data point in the model predictive control in the case of integrating the ZMP and COM projection to LIPM walking robot, in this regard with the current position along x and its reference inputs with similar graph curve as shown small error deviation with center of mass position and velocity. And also, in the figure below (*fig.4.12)* in the other part of the gait also called the left legs can be

gives the following position and velocity curves as they provided in the figure below as shown below, this one also called position of the foot along to the y-direction and its velocity response curves.

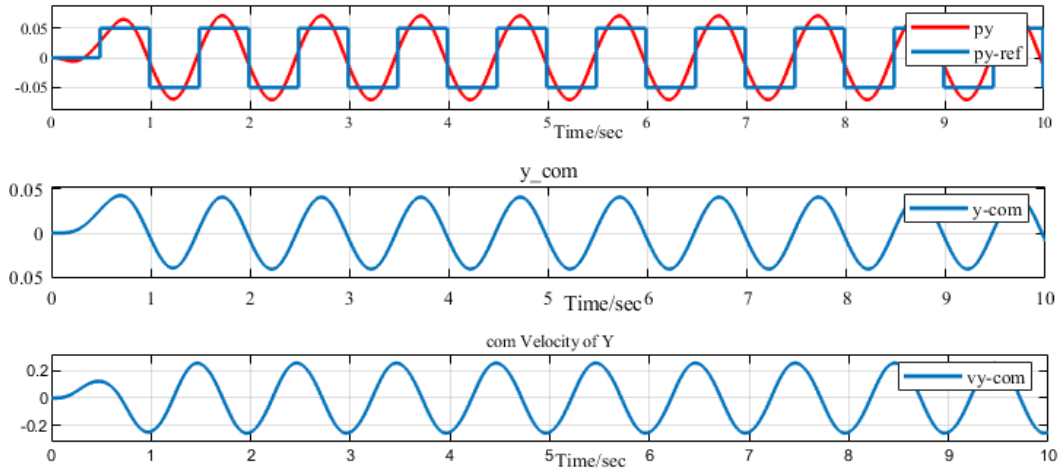


Figure 4.12: MPC generated biped walker left leg data

The design of model predictive control for the walking trajectory generation contained of ZMP and COM projection gives a successful walking output of the walking robot this confirmed by the simulation and the COM trajectory of the measured and the reference gives as and the simulation graph was given by the figure below accordingly.

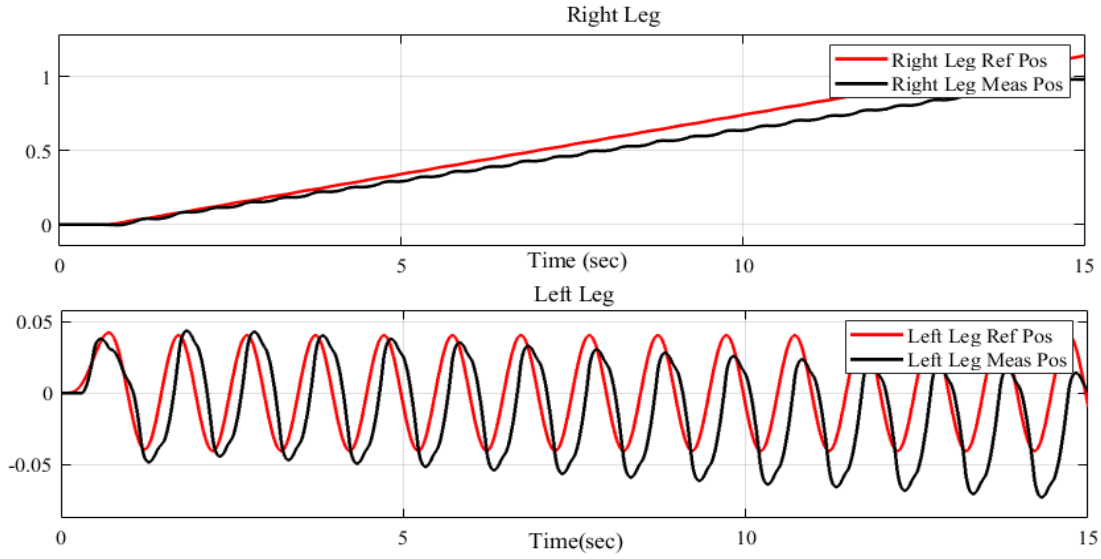


Figure 4.13: COM comparisons of the reference input and output of left and right leg

The center of mass projection along x and y was described in above figure with the measured and the walking robot were compared for the how an MPC controller handles the trajectory for the walking robot and it shows that the results was successful and provides the good outputs with minor deviation compared with the reference and actual one another. The last simulation output of the biped walking robot pattern generation using model predictive

control based on the ZMP and COM trajectory was given below in according to the smooth terrain in the start to the goal point.

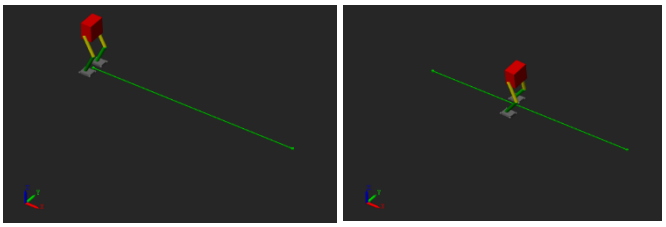


Figure 4.14: MPC biped walking trajectory from the start to mid-point

As you see in the above diagram the MPC scheme of the biped walking robot from its start point was given and thus was showing as were the robot starts walking into the destination point from the given initial point and the middle trajectory point was given below in the *figure 4.15*. The one showing that the robot trajectory in the center of path point to the destination of the goal point. And finally, there was shown that, the biped walking robot meets the destination points to the end point or just the goal points as shown in the *figure.15* below.

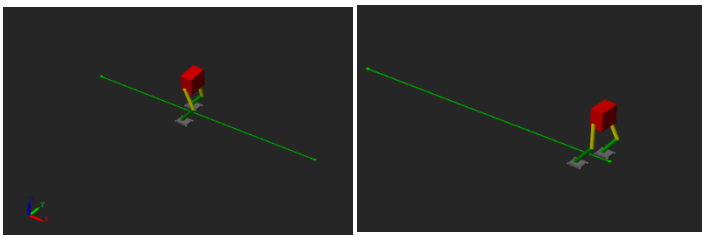


Figure 4.15: MPC biped walking trajectory from the mid to goal-point

Generally the simulation of Dinkinesh humanoid robots in the MATLAB Simulink environment provided with the simMechanics explorer to show an animated simulation for the biped walking trajectory generation using model predictive controller with a trajectory generation, with the application of two key elements of biped humanoid walking in stability; the ZMP and COM projection gives successful output with aid of physical system animation and graphical results in the MATLAB Simulink environments and the walking results give the desired outputs of stable walking in the Animated environment.

4.2 Designing of the path planning system for biped walking robot.

In this scope, I had design the path planning for the biped walking humanoid robot, the simulation comprises two subfield, the first part was the path planning system this is contained of a MATLAB Simulink environment in which the biped walking robot modeled based on the model predictive controlled of doble linear inverted pendulum based on the

path of trajectory that the robot must be walking through and the second section was the path generation in which a biped follow those path based on coordinate programming sometimes this was described as depth image, this path generation system which allow the simulated results based on the MATLAB environment and it was discussed accordingly in the section given below.

4.2.1 Designing of the 3D based biped humanoid robot path planning system.

The path planning system in this thesis designed based on model predictive control in the walking platforms of different path of waypoints. This robot has different parameters compared to the original robot model, to show the robustness of this control with a more realistic robot a thinner torso, smaller feet, less world damping and etc. In this scenario the three-dimensional linear inverted pendulum was used to model the walking representation and the model predictive control for the path planning tasks in the smooth environment based on three types of trajectories of square, circle and star in the three-dimensional coordinate plane. The following dynamical simulation the path planning of walking trajectory based on different trajectory types was given by the following figure below.

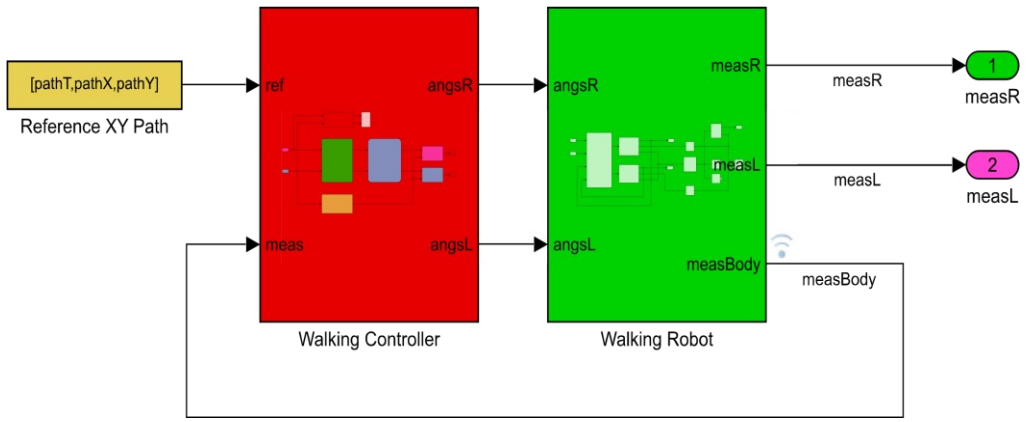


Figure 4.16: Walking robot ZMP control in 3D

The whole system of the trajectory of path in which a walking robot was designed with a walking controller with the help of the input reference of the direction of the XYZ reference path along to the path of x, path of y and path of z with help of zero moment point trajectory generation the simulation of the walking robot based on zero moment point is represented by the figure given below.

As shown in the figure below *figure 4.17*, the ZMP trajectory was designed to model the walking robot based on the zero moment point in which the walking trajectory was going with the help of dynamic stability. The ZMP controller was designed from the reference and

measured position of the foot location and the step logic in which the walking state flow diagram designed for the walking pattern for both legs in term of inverse kinematics problem for both legs, as the output of the combined angles for the plants of the walking robot.

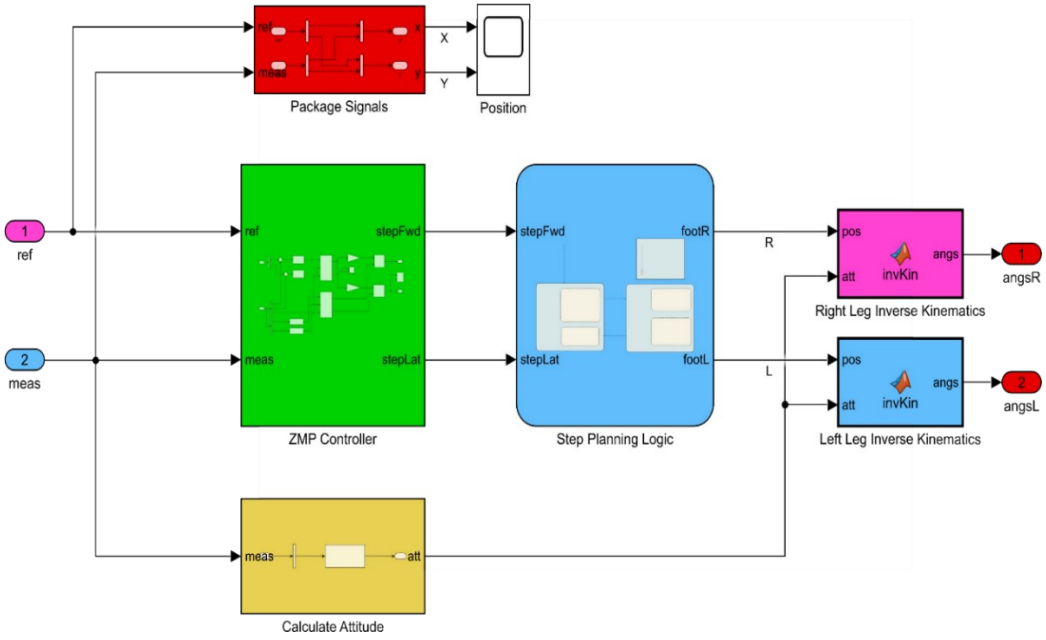


Figure 4.17: Three-dimensional path planning with ZMP trajectory

The initialized step function which designed for the walking transitions for one leg first and the second leg next in the symmetrical periodic phenomena based on the local and global frame reference of the robot in the given walking path trajectory waypoint. In this scope as, I mentioned before in above section, in the section below I had showed the three different path waypoint demonstration based on the square, circular and star path for the robot and it was discussed independently in the section below.

4.2.1.1 Square path planning

As I mentioned the first path trajectory was the square path in which of the robot environment should follow to walk which mean that biped robot follows the square path from start point to the goal point, so that a simulation of the robot start point and goal point, this was achieved by the only ways to follow the path or planning the path with only input reference of the biped walking robot position similar in the coordinate stepping and the walking simulation animation was given below in the *figure 4.18.*

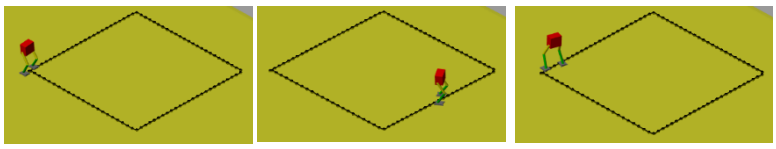


Figure 4.18: The square path trajectory from start points[a], mid-points[b] and goal points[c]

As shown in the *figure 4.18* above the animation of the walking of the biped walking robot in the square path waypoints the BWR start point was obtained accordingly so as to follow the square path with start point fig *4.18(a)* and the middle walking profiles was given in the *4.18(b)*. And finally, the walking path of the biped walking robot based on the square path waypoints was obtained in the square waypoints from start point to the goal point based on the MATLAB Simulink environment accordingly in the figure above *4.18(c)*. In order to compare the zero moment, point of the reference and measured position output the simulation result showed that, the trajectory graphs for a square path for both position and acceleration deviation of the two legs independently from the three-dimensional walking robot presented, as shown in the figure below *(fig 4.19)*. The respective animated simulation of the path planning in the well-known, the square path waypoints was simulated, in this regard the position and acceleration of the animated way points obtained for ZMP location in the support area as well the acceleration results were obtained from the converted local frame of the biped walking humanoid robot.

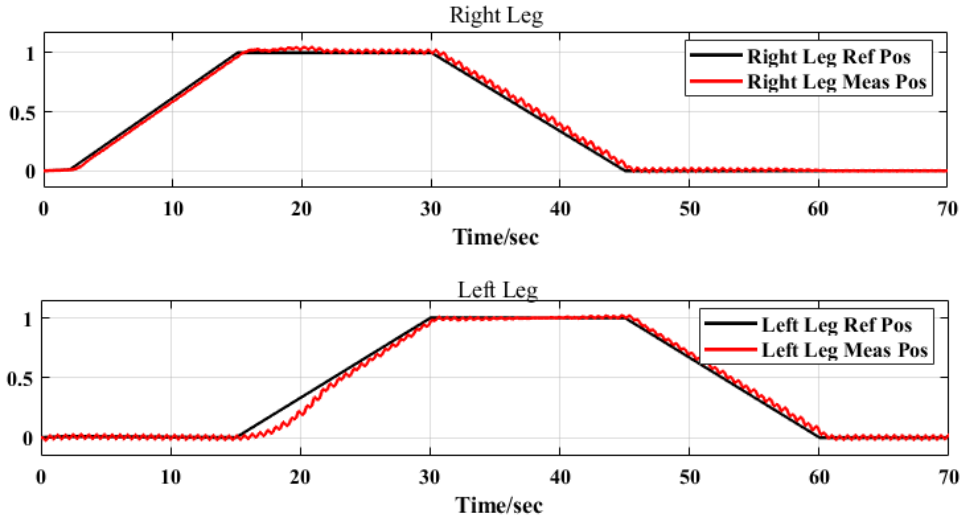


Figure 4.19: 3D square path planning position trajectory curves

In the above *figure 4.19;* the three-dimensional biped walking humanoid robot by using zero moment point based on square path waypoint for position and velocity curves was obtained for both legs in right and left legs curve from the measured and reference waypoint x and y coordinate point and the regarding acceleration curves was described in the figure below.

So, to verify or to show how a robot should walk in the given path, the golden promise of the closed form of the inverse kinematics problem plays an important role for the combined joint angles control toward the foot coordinate frame of the robot and this animated walking trajectory was given below in the *figure 4.20.*

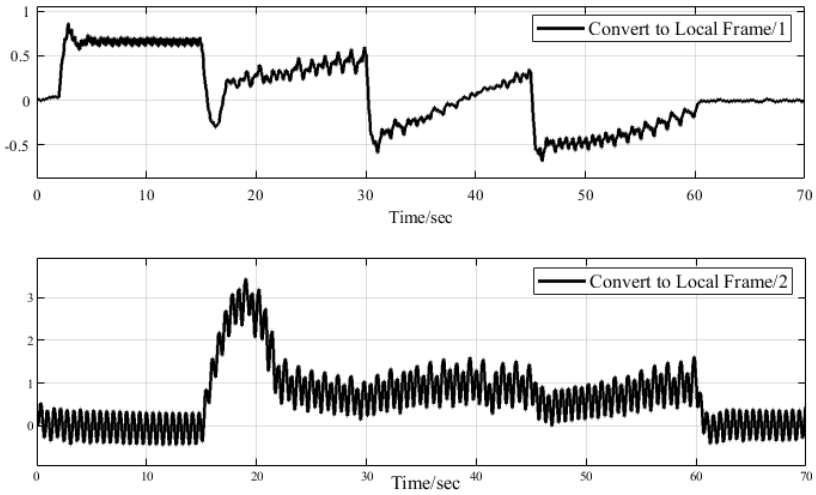


Figure 4.20: 3D square path planning acceleration trajectory curves

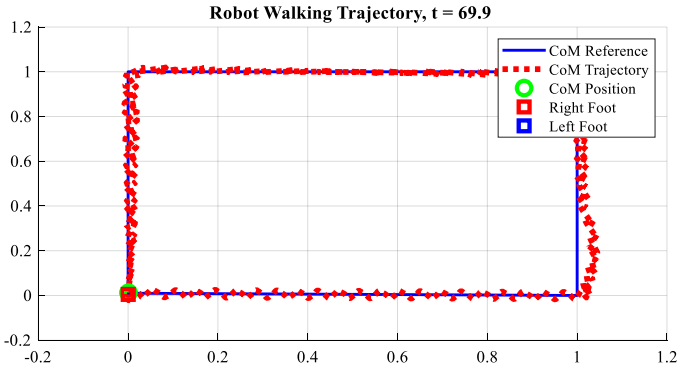


Figure 4.21: Robot waking trajectory in the square path

So as shown in the figure above there was a reference center of mass position with straight line and the center of mass trajectory with dotted line, this shows in the simulation, the walking robot follows a perfect COM position in the center between two legs and the walking robot perfectly follows the start points to the goal point.

4.2.1.2 Circular path planning

As I mentioned the second path trajectory was the circular path of the robot environment which mean the biped robot follow the circular path from start point to the goal point, in simulation the robot start point and goal point was similar and the walking simulation results from the given start point, also given below in the figure.

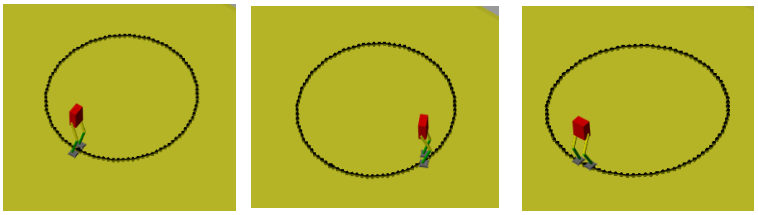


Figure 4.22: The circular path trajectory from start points[a], mid-points[b] and goal points[c]

As shown in the *4.22(a)* above the animation of the walking of the biped walking robot from the start point was obtained accordingly so as to follow the circular path waypoints with start point with and the middle walking profiles was given in the *4.22(b)* above. And finally, the walking path of the biped walking of the robot based on the circular path was obtained in the goal point based on the MATLAB Simulink environment accordingly in the *4.22(c)*. The modeling of the ZMP controller that was designed with MATLAB Simulink environment based on the input reference path for the walking robot, all the input of the ZMP controller was the input reference position of the walking robot. The input reference position was given into the robot for the ZMP path position for the circular path of the walking path and the deviation was explained based on the measured and the input reference ZMP trajectory was obtained from the simulation.

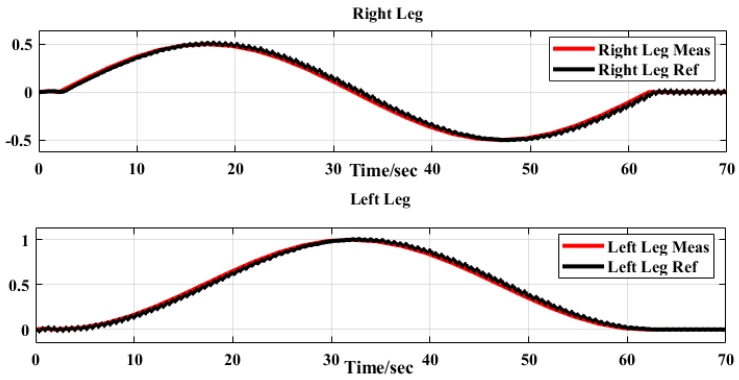


Figure 4.23: 3D circular path planning position trajectory curves

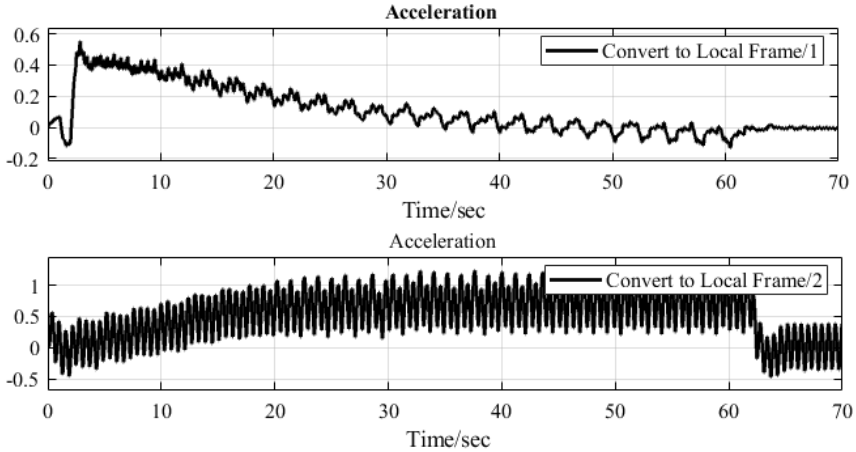


Figure 4.24: 3D circular path planning acceleration trajectory curves

In the above figure 4.23 the three-dimensional biped walking humanoid robot by using zero moment point based on circular path waypoint for position and velocity curves was obtained for both legs in right and left legs curve from the measured and reference waypoint x and y coordinate point and the regarding acceleration curves was described in the figure below. So, to verify or to show how a robot should walk in the given path the golden promise of the

closed form of the inverse kinematics problem plays an important role for the combined joint angles control toward the foot coordinate frame of the robot and this animated walking trajectory is given below in the figure 4.24.

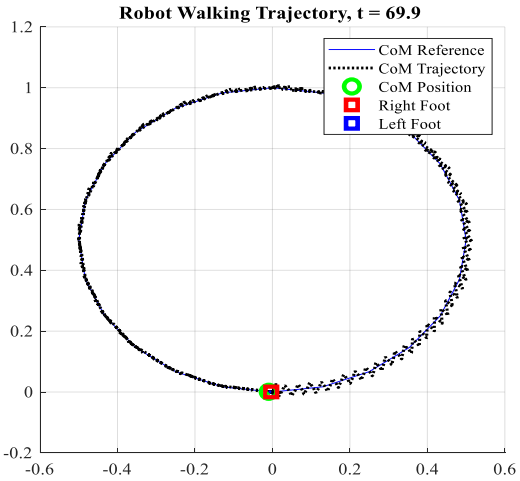


Figure 4.25: Robot waking trajectory in the circular path

So as shown in the *figure 4.25* above there was a reference center of mass position with straight line and the center of mass trajectory with dotted line, this shows in the simulation the walking robot follows a perfect COM position in the center between two legs and the walking robot perfectly follows the start points to the goal point in the useful ways.

4.2.1.3 Star path planning

As I mentioned the third path trajectory was the star path waypoints of the robot environment which mean the biped robot follow the circular path from start point to the goal point, in simulation the robot start point and goal point was the similar and the walking simulation was given below in the figure.

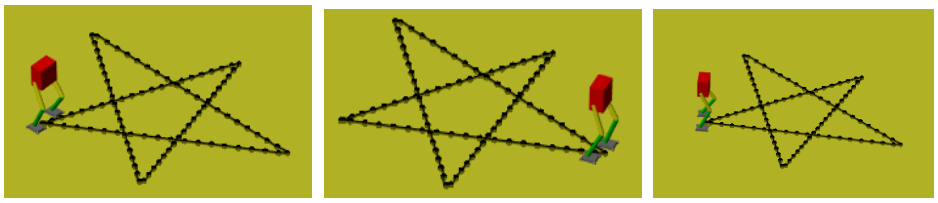


Figure 4.26: The star path trajectory from start[a]. mid[b] and goal points[c]

As shown in the *fig (4.26a)* above the animation of the walking of the biped walking robot was obtained accordingly so as to follow the complicated star path waypoints with start point with and the middle walking profiles was given in the *fig (4.26b).* And finally, the walking path of the biped walking of the robot based on the circular path was obtained in the goal point based on the MATLAB Simulink environment accordingly in the *fig (4.26c).* The modeling of the ZMP controller that was designed with MATLAB Simulink environment

based on the input reference path for the walking robot all the input of the ZMP controller was the input reference position of the walking robot. The input reference position was given into the robot for the ZMP path position for the star path of the walking path and the deviation was explained based on the measured and the input reference ZMP trajectory was obtained from the simulation.

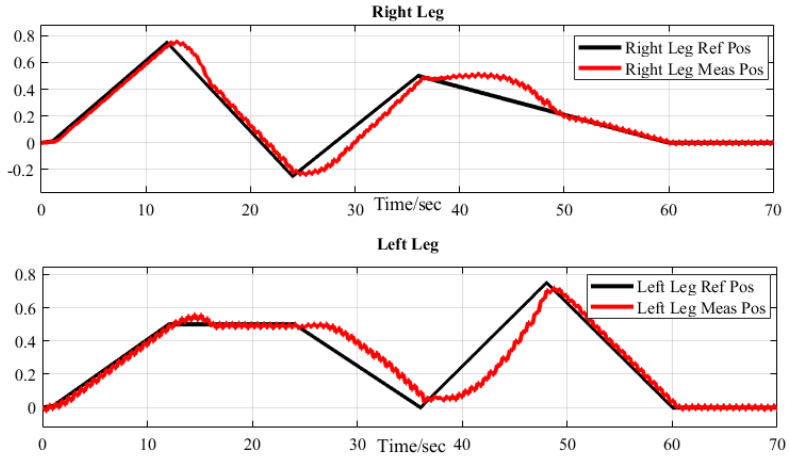


Figure 4.27: 3D the star path planning position trajectory curves

In the above figure 4.27 the three-dimensional biped walking humanoid robot by using zero moment point based on star path waypoint for position and velocity curves was obtained for both legs in right and left legs curve from the measured and reference waypoint x and y coordinate point and the regarding acceleration curves was described in the figure below.

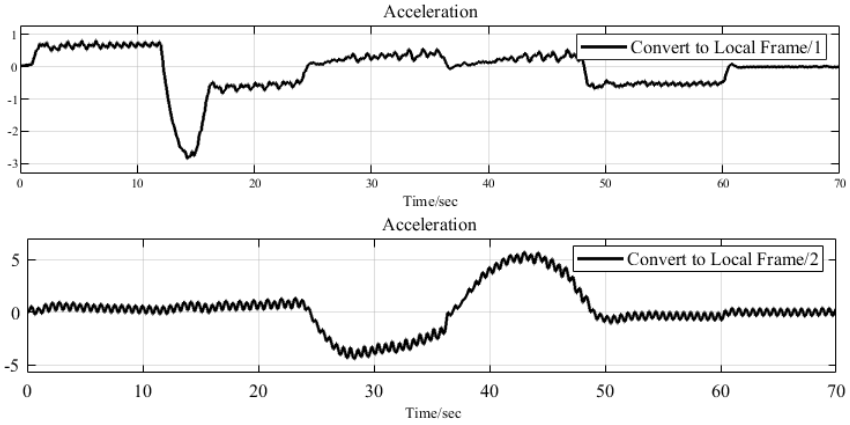


Figure 4.28: 3D the star path planning acceleration trajectory curves

The modeling of the ZMP controller that was designed with MATLAB Simulink environment based on the input reference path for the walking robot all the input of the ZMP controller is the input reference position of the walking robot. The input reference position is given into the robot for the ZMP path position for the star path waypoints of the walking path and the deviation was explained based on the measured and the input reference ZMP trajectory was obtained from the simulation.

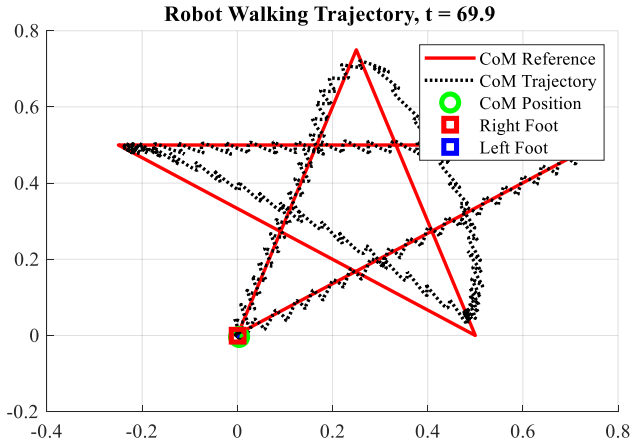


Figure 4.29: Robot waking trajectory in the star path waypoint

The modeling of the ZMP controller that was designed with MATLAB Simulink environment based on the input reference path for the walking robot all the input of the ZMP controller were the input reference position of the walking robot. The input reference position was given into the robot for the ZMP path position for the circular path of the walking path and the deviation was explained based on the measured and the input reference ZMP trajectory was obtained from the simulation.

4.3 Designing artificial intelligence method for biped walking robot

To design the artificial intelligence method for the Dinkinesh biped walking humanoid robot, I had used deep reinforcement learning, which allow the artificial intelligence method without modelled system dynamics, in order to teach the humanoid robot in certain tasks of agents and its environments, since humanoid meaning that it should be aspires the human capability of learning and interacting with its environments accordingly.

So, for this reason I had introduced the one of the machine learning algorithms also called deep reinforcement learning in order to teach the robot to walk without knowing the environment dynamics. By using this reinforcement learning algorithm with an agent called deep deterministic policy Gradient (DDPG), in this section I had made vast discussions on the biped walking robot RL-DDPG with 2D and 3D environment dynamics and by the end of this section I was demonstrated the corresponding simulation results accordingly.

4.3.1 Designing Reinforcement Learning to walking

In this section the designing of reinforcement learning for the biped walking robot presented with reinforcement learning (RL) which is the type of learning that guided by a specific objective in which an agent learns by interacting with an unknown environment, typically in

a try-and-error way. The agent receives the feedback in terms of a reward (or punishment) from the environment; then, it uses this feedback to train itself and collect experience and knowledge about the environment.

Reinforcement learning problems are related to learning which is the best action to perform, situation-by-situation, in order to maximize the aggregated reward. RL agent has to learn a policy (i.e., a complete mapping between situations and actions) by trying actions without any domain expert that has to told it, as in many other forms of machine learning. Another relevant characteristic of a RL problem is that in any situation the agent has to choose between exploiting its current knowledge of the environment (perform an action already tried previously in that situation) or exploring actions never tried before in that situation.

Now in this section, the deep reinforcement learning of biped walking robots was discussed and I had showed how the simulation of a bipedal robot was trained to walk using reinforcement learning specifically deep reinforcement learning which used neural networks with the deep deterministic policy gradient in the MATLAB Simulink environment.

So, let's introduce the problem in the simulation of the walking the robot, what I had defined that the environment, so this was the physical system that I had trying to in some way control to teach Dinkinesh biped robot how to walk. Reinforcement learning comes in in the form of an agent, that we can to think of this as a controller, so what's happening was that the environment was producing some kind of state or observation which would then pass through the agent and it generates an action. It was very important to define an environment for every reinforcement learning problem.

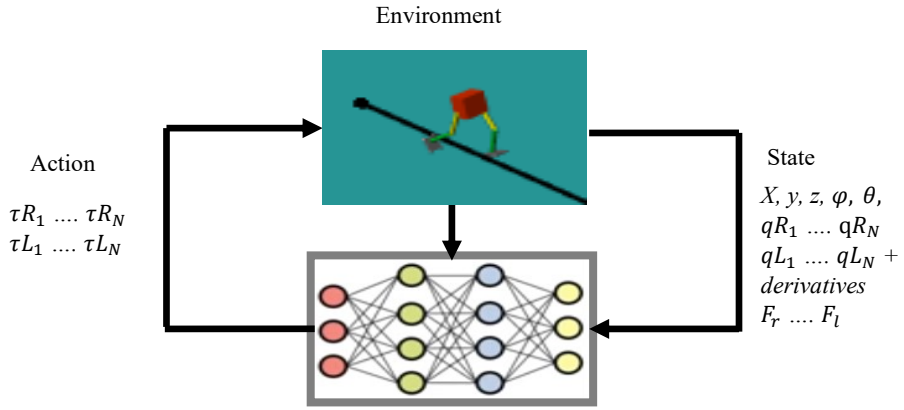


Figure 4.30: Biped walking robot reinforcement learning

Defining an environment meaning to list a set of rules i.e., which action an agent is permitted to select and what states the environment has, what will be the rewards or penalties for the

given sets of rules. As I mentioned previously, for reinforcement learning we have to define an environment. Creating an environment essentially means defining a clear set of rules. What actions is an agent allowed to take in the given environment? What possible states does this environment have? What are the definitions of rewards and penalties? The Environment is the world where an agent or software algorithm can interact or move. The input to the environment is an agent's action taken, current state while the environment's output is the next state and reward. DDPG first of all consists of one deep neural network known as the actor which will effectively act as my controller for the system, by going to accept the state and output an action, so this was what's eventually going to be driving this robot to walk but that's not enough to actually train the entire agent there's actually a second neural network known as the critic and that accepts both the state and the action and outputs an estimated value should require.

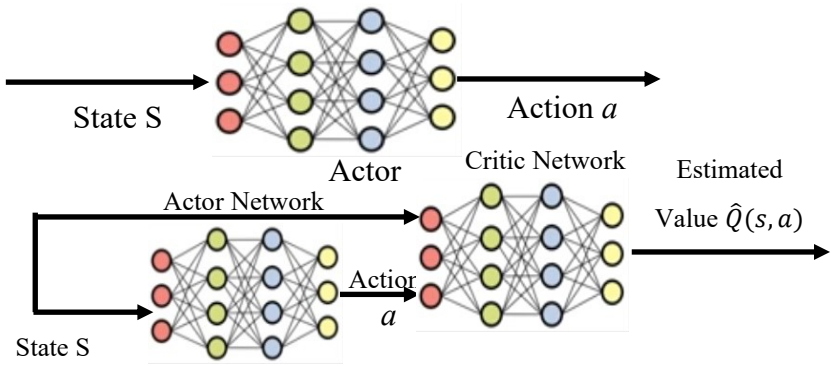


Figure 4.31: RL Problem Actor Critic representation

This particular value, that I had referred to as the Q-function which I can roughly think of as the expected reward, which I had to get from the start at a particular state and that would give a particular action, so when we want to train the critic to correctly estimate how well the biped robot going to walk from any given state chip in a decision now.

REWARD [R(s)]; The reward function *r(s)* returns a numerical value that an agent can get for being in a state after taking a specified action. These rewards tell whether a state is useful or not such that agent gets a higher reward when moving in useful states and lower reward when moving in undesirable states. Reward acts as feedback for an agent which can be positive or negative. Nature or amount of reward that has some effect on actions and sometimes minor changes make a big difference in agent behavior. The cumulative future reward also called return is given as:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \ldots + r_{\infty} \quad (4.10)$$

where subscript *t* denotes a specific time step. Here we need to end a series of rewards instead of infinite returns. It was more interested in episodic tasks and cumulative discounted rewards. Episodic tasks are those which terminates after a certain time step or when the agent reaches in a terminal state.

We may denote the long-term discounted reward as a utility of a state. In this research thesis, based on the third research questions the integration of artificial intelligence with robotics based on the machine learning algorithm the biped walking humanoid robots walking dynamics can be controlled accordingly, in this concern, I went through to designing the reinforcement learning based on deep neural network with continuous space actions with the given environment dynamics. There is a certain block diagram representations in the figure below in which a walking dynamic of balancing and leg and trunk walking trajectory in the dynamic representation was represented in the figure below.

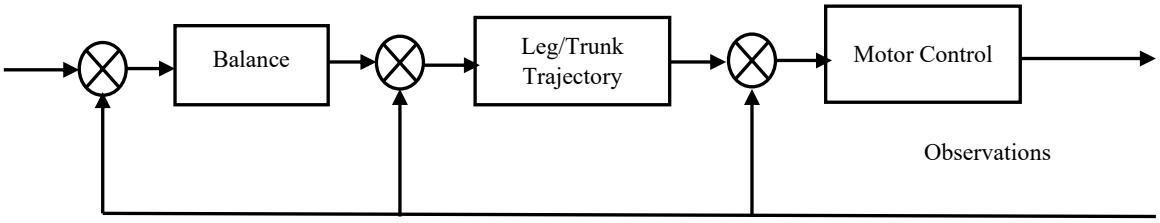


Figure 4.32: a simple walking and balancing control of biped walking humanoid

There are so many ways to control the biped walking humanoid robots either in the form of traditional approach and machine learning algorithm or artificial intelligence method the basic difference between the possible control method that can be represented by the figure below.

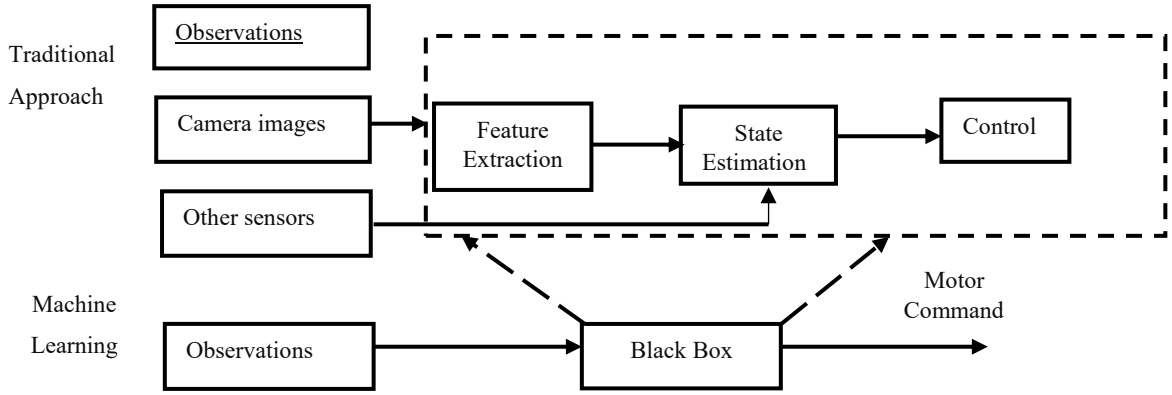


Figure 4.33: Traditional vs Machine Learning method to control the biped walking humanoid robot.

It was very clear that; the question was what type of control method to be applied in the balancing and walking control of the humanoid robots to obtain an optimal constraints or to

maximize the performance index of the cost function in dynamic locomotion, in this regard the machine learning algorithm was just take the prior performance and the point was how to represent the blackbox or the unknown environments based on the reinforcement learning approaches meaning a best sequence of actions that will generates the optimal outcomes that will generate the most reward based on the best actions. As shown in the *(figure 4.34)* below in the figure, the reinforcement learning problem representation was described accordingly in a such ways of what type of the agent or the policies should be represented for the unknown environment for the best sequence of actions that will reward the best optimal outcomes in the given observation or state of the environments.

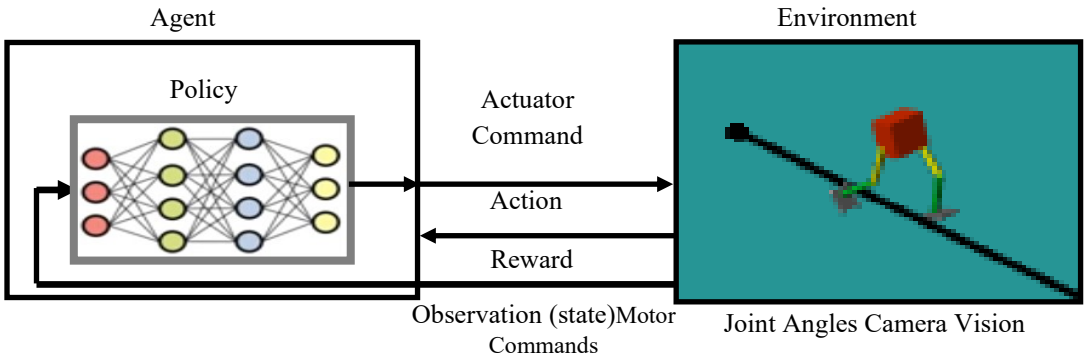


Figure 4.34: RL Problem Representation

The prior assumption in the algorithm certainty was represented by the try and error method and by selection of the right RL learning algorithm that suit with to be applied actions or the application areas of the tasks.

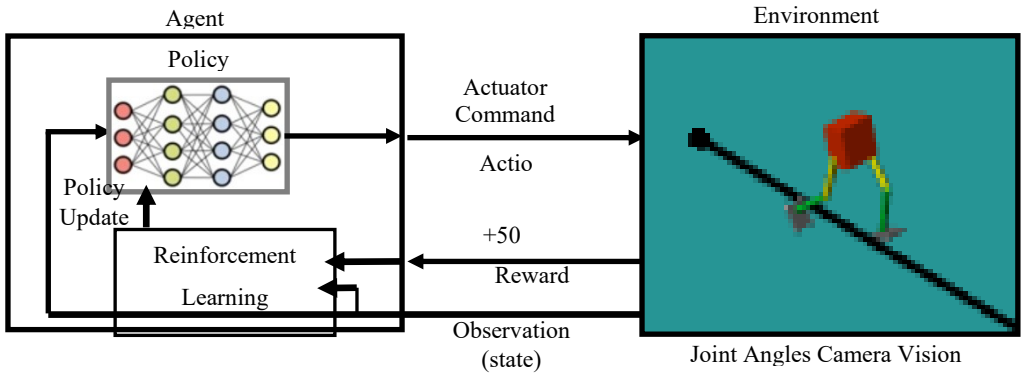


Figure 4.35: Agent and environment interaction based on RL algorithm

In the above *figure 4.35* shown in the block diagram the value which is the total expected reward from that state observation and from its onwards into the future in which instantaneous benefit of being in a specific state for the promise of future high reward might not be the best action because of the present reward is always greater than the previous reward and the future reward are not as reliable thus of the RL discounts the future rewards.

For these important understandings of how an agent interacts to an environment dynamic, there are two topics of in the policy updates in the RL agents this were an exploration and exploitation which mean that an agent must be explore a new area in the environments and collecting the most reward we already know about that because of tricky to balance, so this is all about analogous to the control problem as given in the *figure 4.36* below.

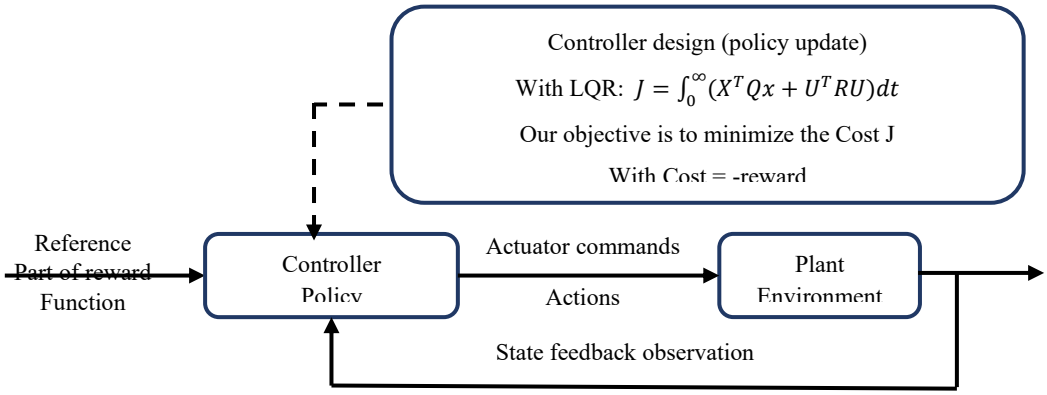


Figure 4.36: RL analogous to the control problem

In the sense of the control theory there is a reference input to the controller which is replaced by the part of the reward function which is subjected to the policy in which or what types of actions to be performed in the plant or in the environment which compared to the reference input through state feedback, what we call it state observation in the RL agent. So, the entire process of computing control in the advancement of reinforcement learning process is clearly given below the block diagram below.

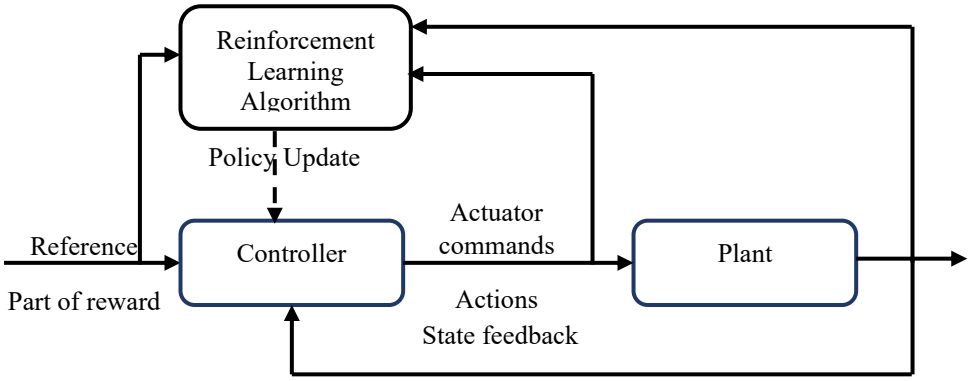


Figure 4.37: Reinforcement Learning analogous to the control process of the biped walker

The basic concept and advancement of the reinforcement learning was it can be possible to design the controller without knowing the information of the plant dynamics because of it only mimics the representation of the plant in terms of environment dynamics, here the questions was how to design the controller without knowing the information of the plant with the plausible machine learning algorithm is by selecting the reinforcement learning

algorithm for this reason we need to understand our system is that choosing RL is a good option or not if we chose RL: we should have to set up the controller structure what we call it (policy) and we have to know what success is just (reward) and we have to learn efficiently the RL algorithm we wants to select.

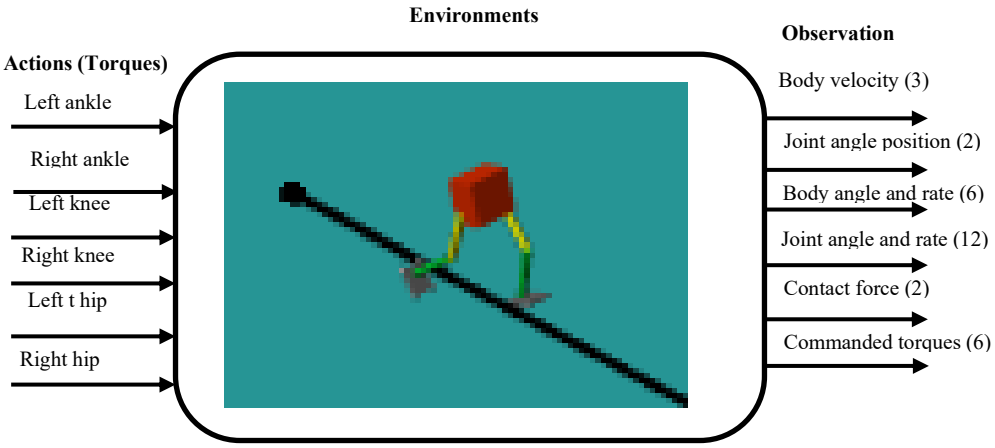


Figure 4.38: Actions vs observation maps of the RL walking problem

I had defined action versus observation maps of the RL walking problem in the Dinkinesh walking humanoid robot, for the design based on the actions of the specific topics of torque command of both ankle, knee, hip and the output observation of body velocity, joint angle position, body angle and rate, contact force and the commanded torques. So, in all a control system should takes all the 31 observations and calculate the values of the six motor torques simultaneously. So, in the traditional control system logic, loops, controllers, parameters and etc. So, this how a traditional control system looks so complex even for simple tasks so in this simple scope of the traditional control system replaced by the new approach of machine learning algorithm called the reinforcement learning agent. With one actor which maps those 31 observations of six motor torque commands of six actions and a critic network to make a training and more efficient the learning to be smarter. We want the body of the robot to walk forward in this case instead of using distance we use the forward velocity in those ways the desired the robot to walk fastly rather than slower by training the reward function by the reward of the forward velocity by

$$\text{Reward} = V_x \tag{4.11}$$

Rt = Vx in this case the robot doesn't walk efficiency for this reason we going to add the penalty function which reduce the sudden diving by

$$\text{Reward} = V_x + 0.0625 \tag{4.12}$$

Vx is the forward velocity and 0.0625 is the time step reward. Then we going to show some hopes to walks it jumps like frog but it is not what we should want to do for in this regard I going to add the initial height which keep robot at walking height as possible.

$$\text{Reward} = V_x + 0.0625 - 50z^2 \tag{4.13}$$

But still, it is not showing a natural walking to going through again we going introduce a reward with minimize actuator effort.

$$\text{Reward} = V_x + 0.0625 - 50z^2 - 0.02\sum_i\left(u_{t-1}^i\right)^2 \tag{4.14}$$

Now we can find some similar walking like human walking but it is not efficient in the optimal ways, in the following the given path now we going to reward by don't stray from path by introducing reward.

$$\text{Reward} = V_x + 0.0625 - 50z^2 - 0.02\sum_i\left(u_{t-1}^i\right)^2 - 3y^2 \tag{4.15}$$

So generally, in RL agent actor maps the 31 observations to the six action which shows what to do from the given input observation and decide the action which is also called the policies and finally the critic is the learning algorithm which allows and decide a parameter in the actor.

4.3.1.1 2D walking robot reinforcement learning

To model the walking action of biped humanoid robot based on the reinforcement learning algorithm I had designed two MATLAB Simulink model on with RL problem based on the two-dimensional biped walking robot and the other contained of the three-dimensional walking robot by creating a deep deterministic policy gradient neural network with training agent for both case and in this section the 2D walking robot reinforcement learning problem. And the MATLAB Simulink model is just given below in the *figure 4.39.*

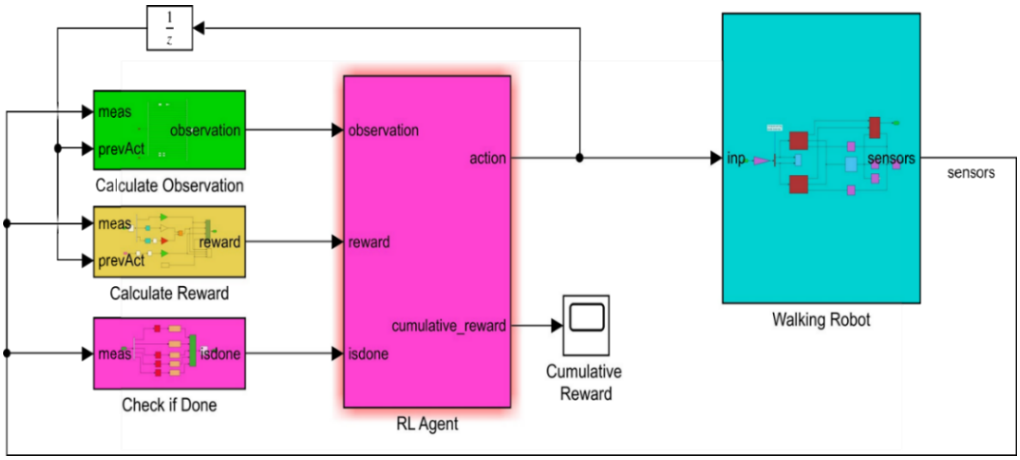


Figure 4.39: Simulink model of 2D biped walking robot based on RL problem

In the above figure the plant of walking robot was an environment which utilize a six-torque command action from the 31 observation of plant dynamics this action will decided by the reinforcement learning agent by calculating the possible 31 observation and provides its own reward this was made by the entire RL problem formulations in the simulation environments. For this regard let's to see how the reward function itself can be represented in the MATLAB Simulink environment so as to make the walking platform stable and dynamic and the followed by the *figure 4.40*, which contained of the reward function for the 2D biped walking humanoid robot.

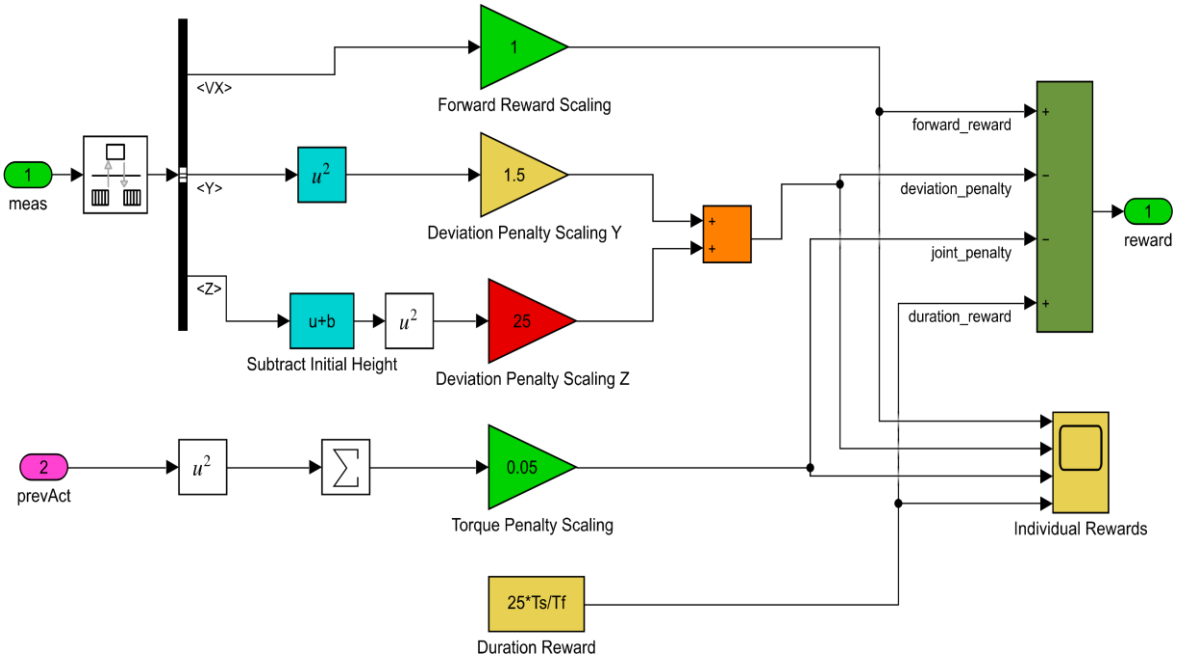


Figure 4.40: Simulink model of 2D reward function BWR based on RL problem

So, I'll run through the reward signal and you see that it was inspired by the literature so what does it mean for us to walk well as shown in the *figure 4.40,* so the first thing we'll do was we'll by adding a reward with the parameters of reward velocity in the forward x-direction which means moving forward in a straight line but that's was not enough to make sure the robot doesn't deviate too far from the line or fall down which means that we're gonna apply a penalty on the y and z dimension displacement to make sure that, the robot was on track besides that another thing that we need had to do was to put some penalty on the joint effort or the joint torques that were applying and the reason is that if we don't penalize this then output get some aggressive motions not sure they will make the robot move quickly but may not yield very realistic physical results, the last part in the reward was actually what that called a duration reward as a survival reward and this was preventing a very common local minimum. Now the robot very early on learns to fall forward and know immediately maximize its forward movement, reward but we needed to get around that by adding some

kind of you known reward which allows to boost for actually surviving longer without falling so that was my final reward function as shown in the figure 4.40 above. Hereby simulating the walking robot in MATLAB Simulink environment in the two-dimensional environment, the best optimal result can be achieved by the frequency of training of the agents with trial and error, for this reason I had simulated four training agents to the action of the walking of the biped humanoid robot, so the following topics can be discussed by the section below.

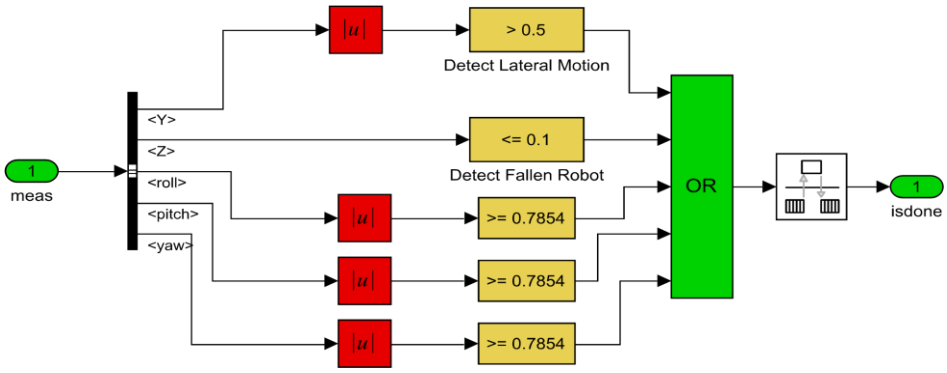


Figure 4.41: 2D RL Loss Measurement

This reinforcement learning problem as described in the *figure 4.40* contained of the observation, reward and isdone block in similar ways the measured data from the observation through exploration and exploiting just compared with the reward provided by the reward function. As shown in the *figure 4.42* those measured data from the plant of the walking robot, in the left leg and right leg with two-dimensional biped walking robot those measured data just perceived from the sensor just packed in the MATLAB Simulink library like a force sensor.

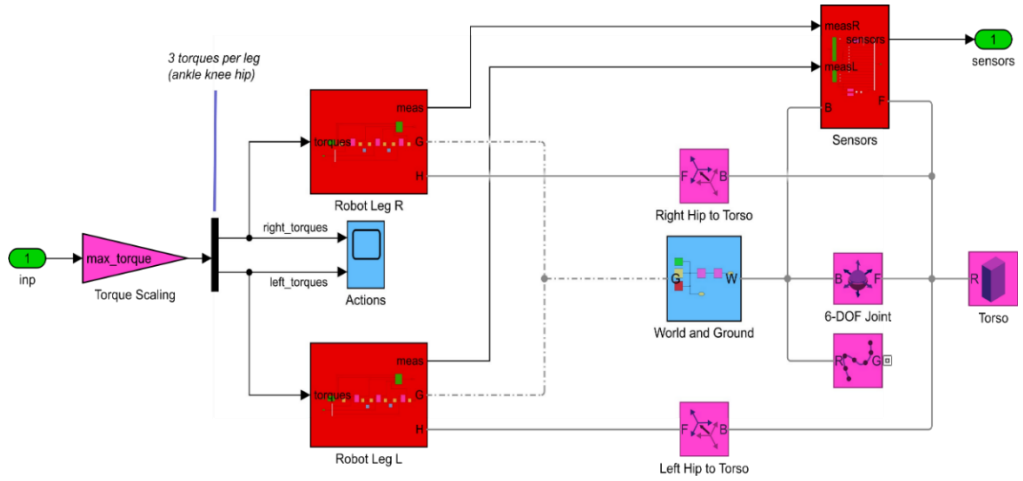


Figure 4.42: 2D RL Biped Walking Robot Plant

The entire walking bipedal humanoid robot plant dynamics which contained of the twelve degree of freedom legs which contained of the MATLAB Simulink with the simscape MATLAB physics library were as presented in the figure above *figure 4.42.*

4.3.1.1.1 2D walking robot the first training RL agent

In the reinforcement learning problem formulation, the required output of the action well performed based on the more frequent training of the RL agent, which mean that the agent is an expert which tells the system how to act in the unknown environment dynamics for specified actions. So, the reward function provides the combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling that provided by the duration reward all the induvial reward was given below in the figure as the follows for the first trained agents.

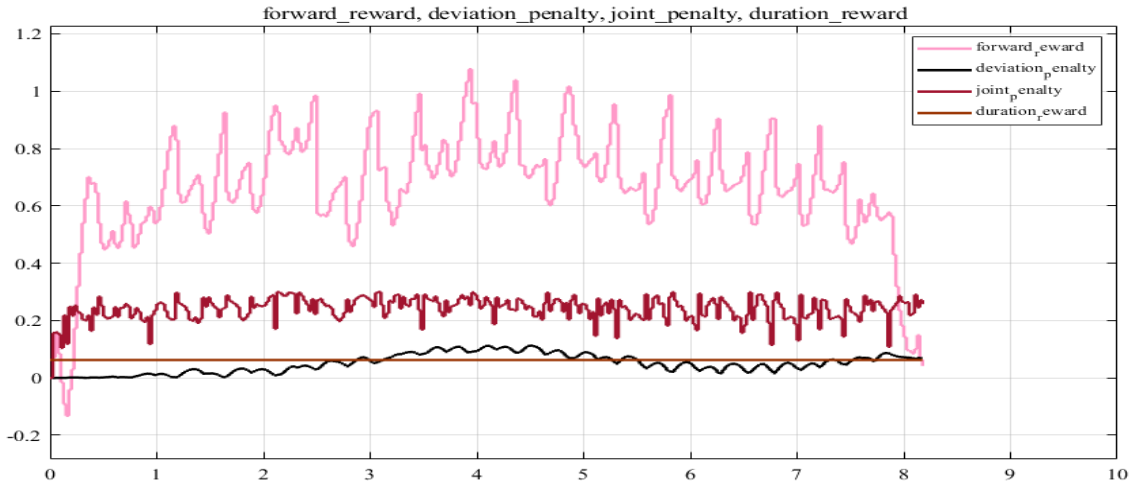


Figure 4.43: The 2D Induvial Reward for the first trained agent of the RL problem.

As shown in the figure above there are four reward curves that was the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space.

So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adapt the unknown environment dynamics and in order to responds the above figures for the possible action of walking in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking for the torque command as the figure shown below.

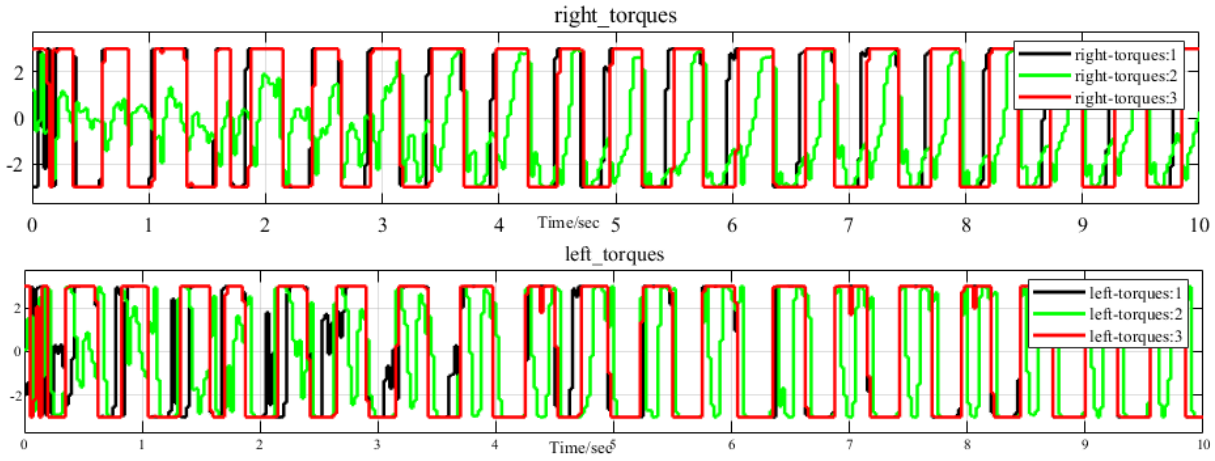


Figure 4.44: RL sequence of action for the first trial of trained agent

So, the above motor torques command sequence of action in which a motor torques for the left and right legs which provides the possible outcomes of optimal reward as the given above in the previous figure the general cumulative reward also provides in the figure below.

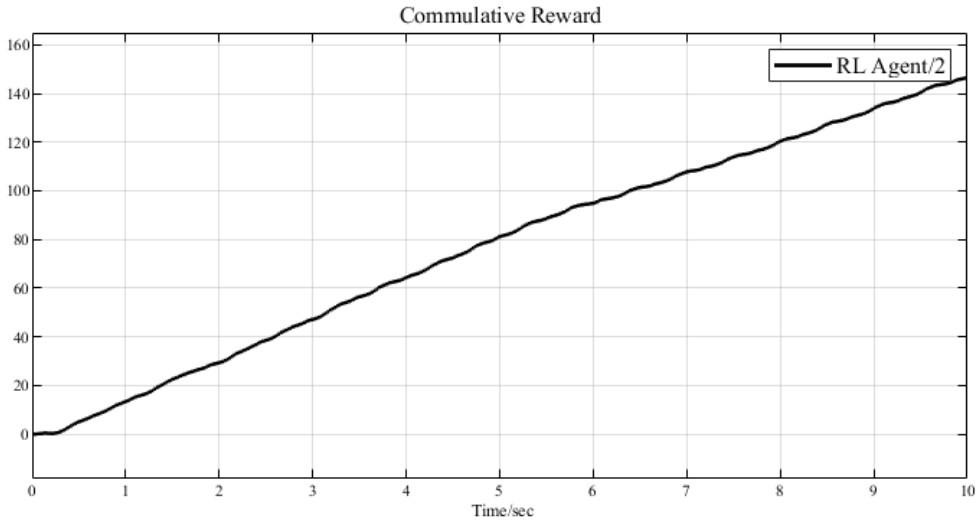


Figure 4.45: The 2D RL first trained agent cumulative reward

To show how the first trained agent behaves in the walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the *figure 4.46* below.

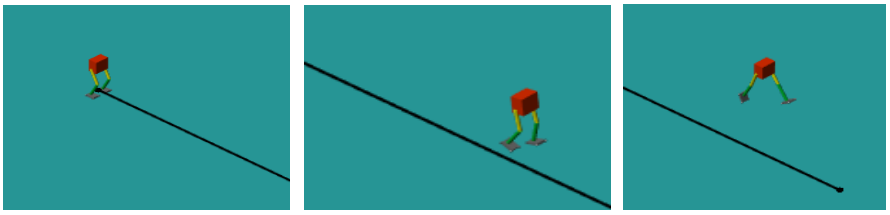


Figure 4.46: 2D walking robot the first training RL agent simulation result[a][b][c]

As we see in the above *figure 4.46* the walking robot the first trained agent based on the RL in the four figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics.

4.3.1.1.2 2D walking robot the second training RL agent

So, the reward function provides the combination of forward velocity reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward was given below in the figure as the follows for the second different values trained agents.

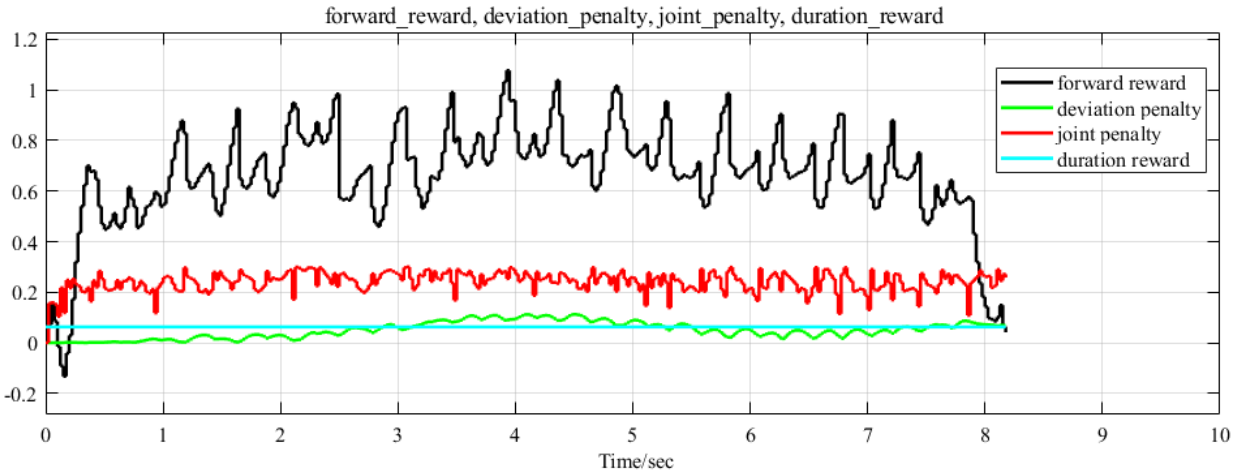


Figure 4.47: The Induvial Reward for the second trained agent of the RL problem

The forward velocity reward which allows the biped humanoid robot to walk in through forward direction, the velocity was just decide how fast the robot to walk from the start point to the goal point and how the robot deviate from the path of walking this one determined from the deviation penalty and the duration reward was so important when the robot walks how fast as well as how much time it requires to keep the robot path from the deviation of path.

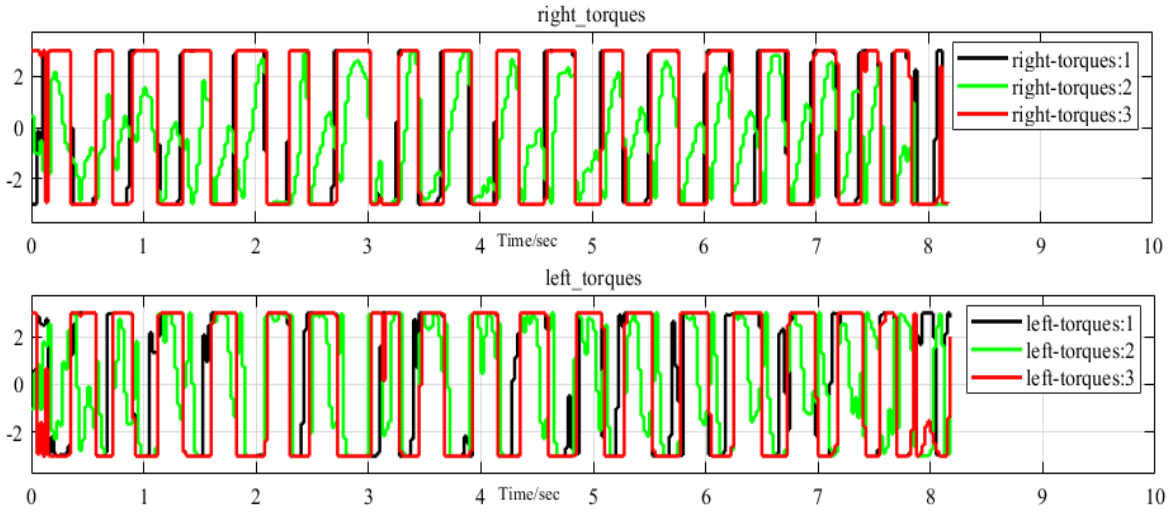


Figure 4.48: RL sequence of action for the second trial of trained agent

So, my objective according to the simulation in the second trained agent responds the above figures for the possible action of walking in which the induvial reward of the forward reward, deviation penalty, joint penalty and the duration reward for the possible sequence of action in the walking for the torque command as the figure 4.48 shown above.

So, the above figure depicts that the motor torques of each legs sequence of action in which a motor torques for the left and right legs which provides the possible outcomes of optimal reward as the given above in the previous figure the general cumulative reward also provides in the *figure 4.49* below. The fundamental basis for the deep reinforcement learning with the deep deterministic policy gradient the combination of the reward that provided by the reward function that is the sum of forward reward, deviation penalty, joint penalty and duration reward just also called the cumulative reward that was presented in the *figure 4.49.*

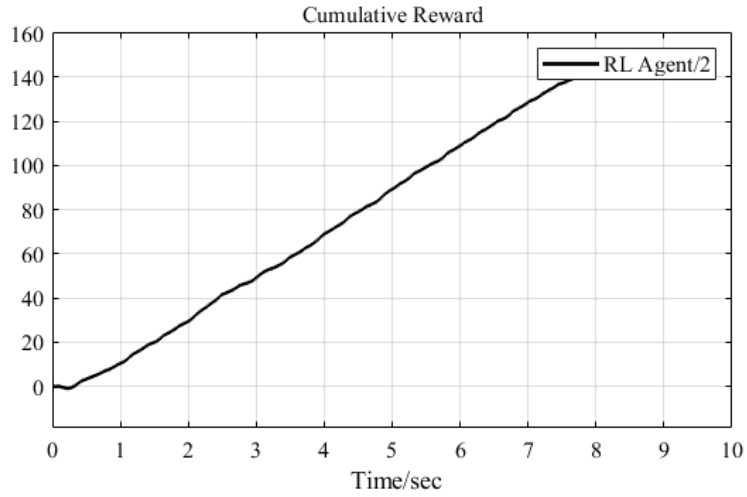


Figure 4.49: The 2D RL second trained agent cumulative reward

To show how the second trained agent behaves in the walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the figure below.

Figure 4.50: 2D walking robot the second training RL agent simulation result[a][b][c][d]

As we see in the above figure the walking robot the second trained agent based on the RL in the four figures marked [a] to [d] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics through the animation.

4.3.1.1.3 2D walking robot for the third training RL agent

So, our reward function provides the combination of forward velocity reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward is given below in the figure as the follows for the third different values trained agents. So, my objective according to the simulation for the third trained agent responds the above *figure 4.51* for the possible action of walking in which the induvial reward of the forward reward, deviation penalty, joint penalty and the duration reward for the possible sequence of action in the walking for the torque command as the *figure 4.52* shown below.

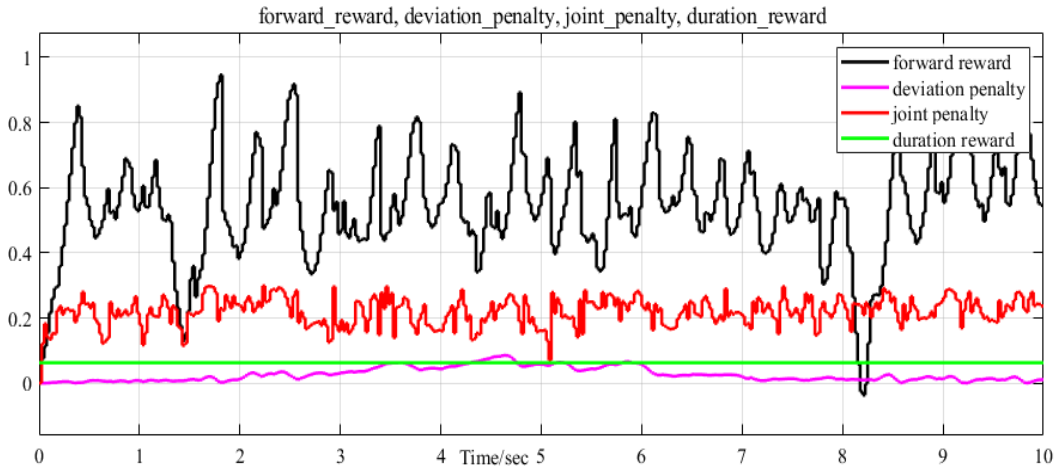


Figure 4.51: The Induvial Reward for the third trained agent of the RL problem

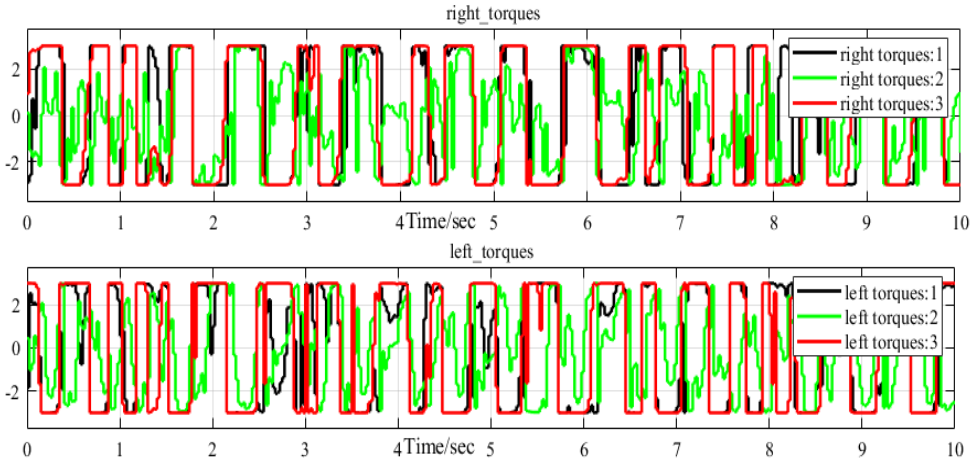


Figure 4.52: RL sequence of action for the third trial of trained agent

So, the above motor torques sequence of action in which a motor torques for the left and right legs which provides the possible outcomes of optimal reward as the given above in the previous figure the general cumulative reward also provides in the figure below.

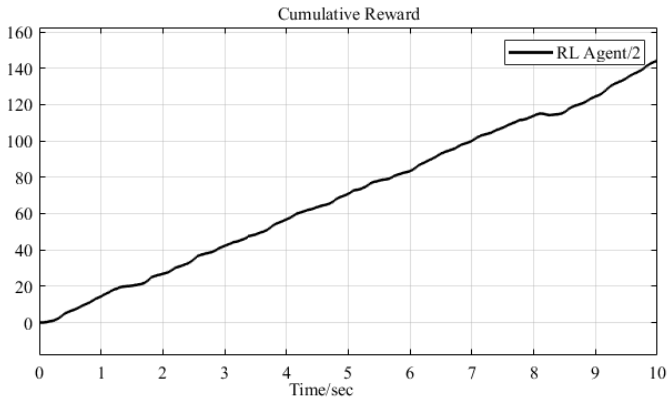


Figure 4.53: The 2D RL third trained agent cumulative reward

To show how the third trained agent behaves in the walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the *figure 4.54* below.

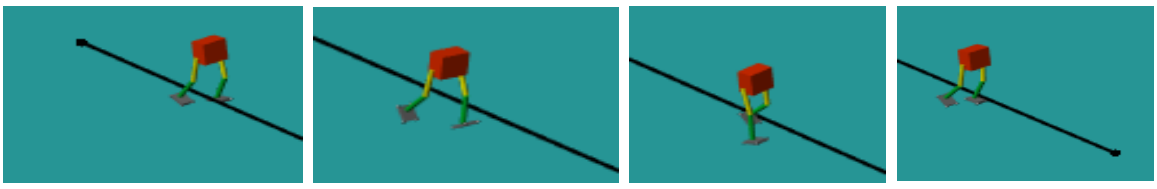


Figure 4.54: 2D walking robot the third training RL agent simulation result[a][b][c][d]

As we see in the above *figure 4.54* the walking robot for the third trained agent based on the RL in the four figures marked [a] to [d] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. So as shown in the figure 4.54 the first diagram of the figure when the robot walking from

the initial path to the final path goal points a to d when the robot to walk from the initial points to the final control goal to desired reached out there.

4.3.1.1.4 2D walking robot the fourth training RL agent

So, our reward function provides the combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward is given below in the figure as the follows for the second different values trained agents.

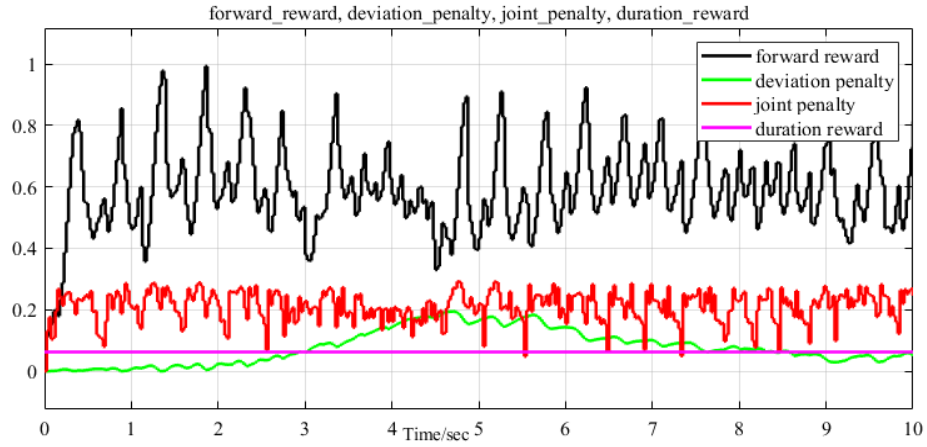


Figure 4.55: The Induvial Reward for the fourth trained agent of the RL problem

So, my objective according to the simulation the first trained agent responds the above figures for the possible action of walking in which the induvial reward of the forward reward, deviation penalty, joint penalty and the duration reward for the possible sequence of action in the walking for the torque command as the figure shown below.

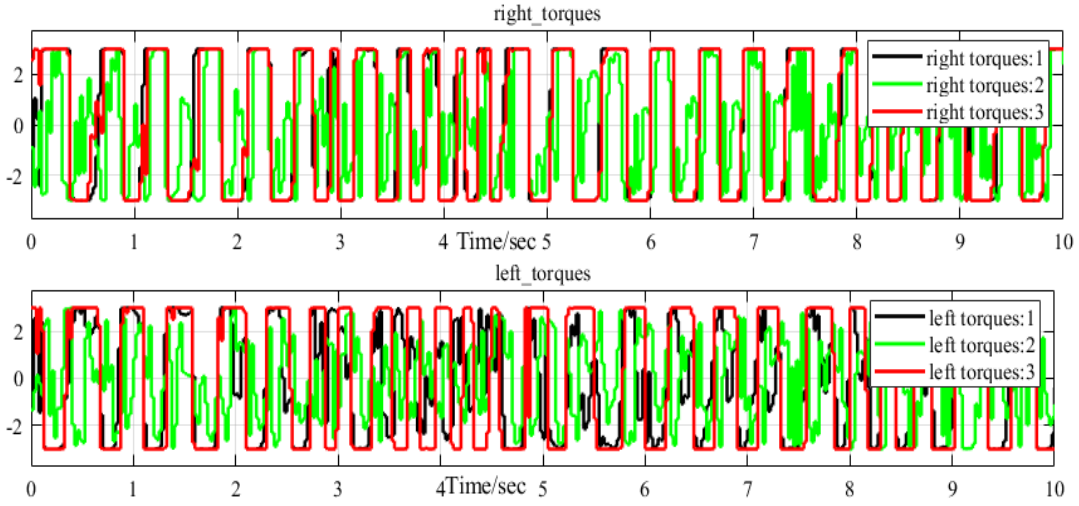


Figure 4.56: RL sequence of action for the fourth trial of trained agent

So, the above motor torques sequence of action in which a motor torques for the left and right legs which provides the possible outcomes of optimal reward as the given above in the previous figure the general cumulative reward also provides in the figure below.

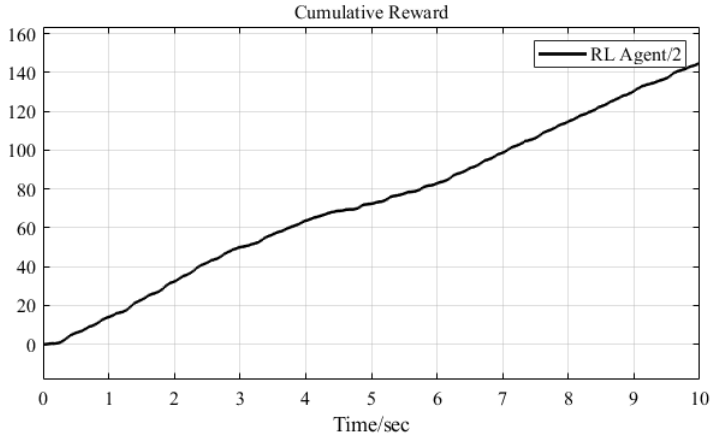


Figure 4.57: The 2D RL fourth trained agent cumulative reward

To show how the fourth trained agent behaves in the walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the fourth trained agent in the figure provided the *figure 4.58* as shown in the figure below.

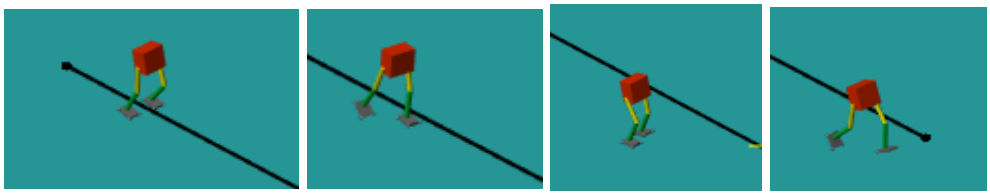


Figure 4.58: 2D walking robot the fourth training RL agent simulation result

4.3.1.2 3D walking robot reinforcement learning

In the above section I had discussed the 2D biped walking robot with deployment of reinforcement learning with deep deterministic policy gradient learning agent and thus shows a good output in the best walking animation outputs. In this section the design of the reinforcement learning for the three-dimensional(3D) walking robot in the three-dimensional walking environment in which a robot to walk in the forward velocity of the reward function plus the other optimizing parameters. Similarly based on the reinforcement learning algorithm the MATLAB Simulink model on with RL problem based on the three-dimensional biped walking robot by creating a deep deterministic policy gradient neural network with training agent for the five trial and error response. The difference between the 2D and the 3D biped walking humanoid with RL was, the first used only three joint actuators for each leg and the 3D used 5 to 6 joint actuators for each leg, this were having a capability of creating the model deference among the model dynamics, the corresponding simulation

of 3D biped walking humanoid robot based on reinforcement learning the MATLAB Simulink model designed for the walking robot which comprises of the walking robot plant and the RL agent represented by the figure given below with RL agent and its environment interaction by calculating observation, the reward and compare its action and rewards to maximize reward and minimize the loss.

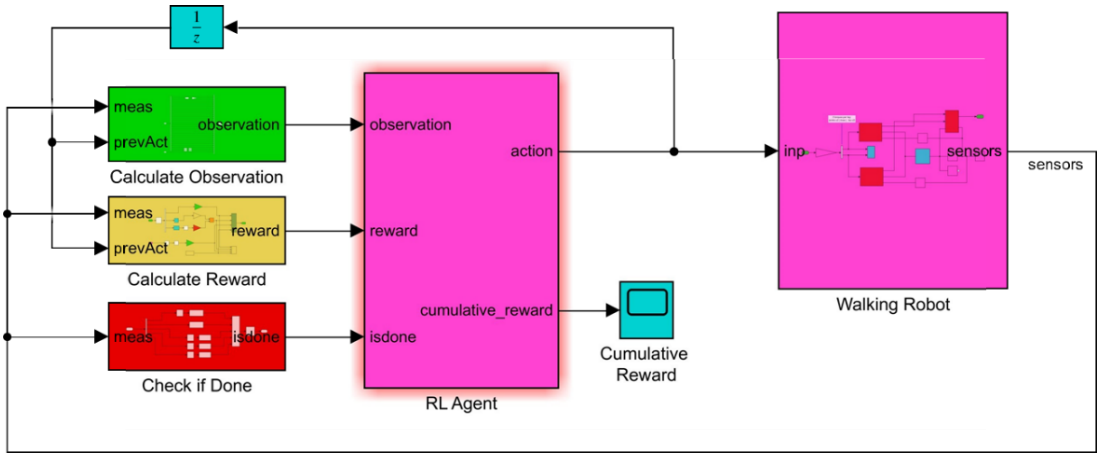


Figure 4.59: Simulink model of 3D biped walking robot based on RL problem

In the figure 4.59 below the plant of walking robot plant which contained of an environment which utilize a six-torque command action from the 51 observation of plant dynamics this action was decided by the reinforcement learning agent by calculating the possible 51 observation and provides its own reward this was made by the entire RL problem formulations in the simulation environments. So, actually how this works, first the sensors perceive an information about the state information in the unknown environment dynamics what just called an observation, this observation can be obtained through deep exploring and exploiting, after an observation were calculated that generate an action, thus action which it gets a reward or punishment through reward function.

For this regard, I had showed how the reward function itself can be represented in the MATLAB Simulink environment, so as to make the walking platform stable and dynamic as the respective Simulink block by the following figure which contained of the reward function that was given below in the following in the figure, fig (4.60).

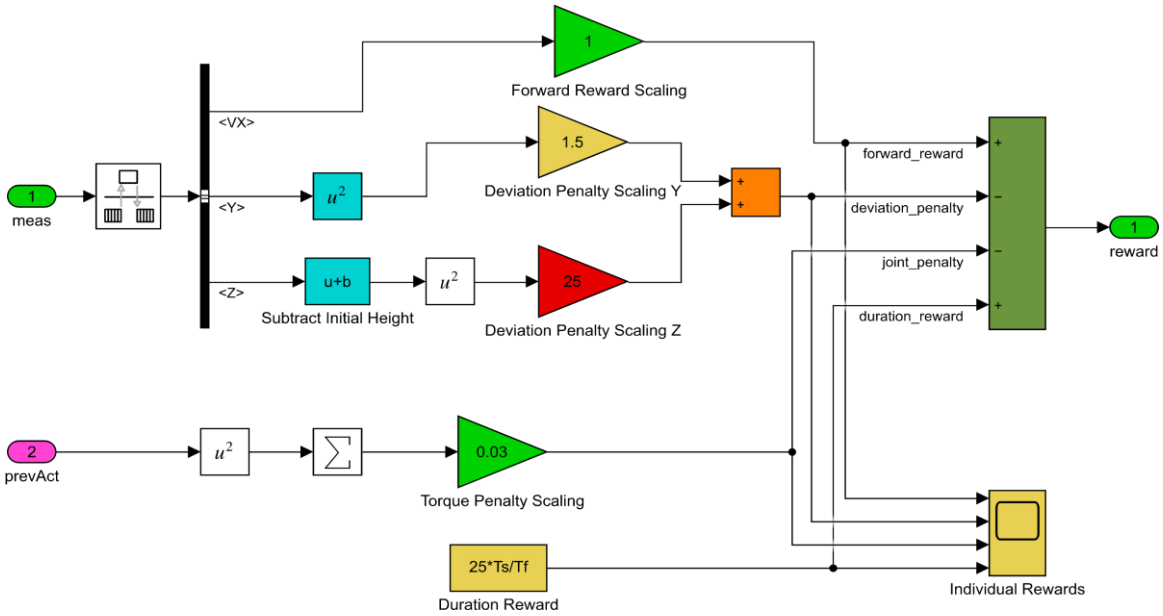


Figure 4.60: Simulink model of 3D reward function BWR based on RL problem

In the figure above *(figure 4.60)* the heart of RL agent called the reward function was given below in which the measured environmental observation in which its rewards for the specific action like thus to set a certain kind of torque of motor in the required state of action, like the forward reward scaling with the gain of one, and the deviation penalty scaling along y and along to z with the torque penalty scaling and finally the duration reward and an RL agent provide what actions to be applied to the walking biped robot, since what was just an observed state which describe an specified action to be performed by loss error which prevent the biped walking from deviate to fall, as just like the error function in the classical control theory. This error loss should be measured and it must sate up, in order to keep the robot from deviating to fall over. This function which calculates the error in which the deviation for the particular state action from the given observation were provided in the figure given below.

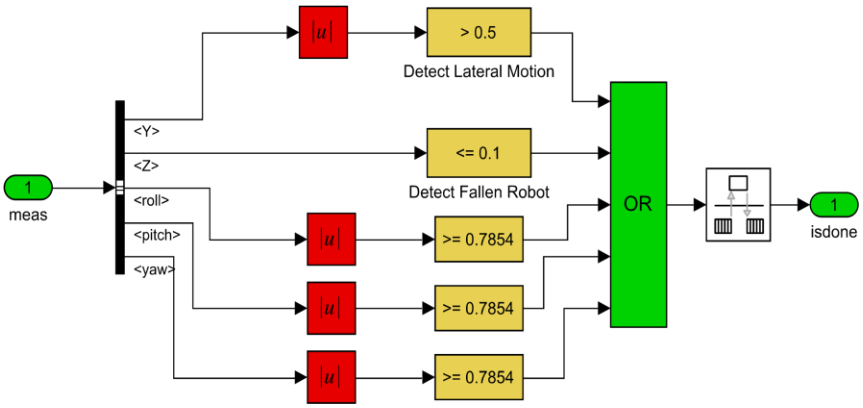


Figure 4.61: 3D-RL Error loss measurement

4.3.1.2.1 3D walking robot the first training RL agent

In this section, the first trial to train the 3D walking biped humanoid robot, in the RL problem formulation, the required output of the action well performed based on the more frequent training of the RL agent, which mean that the agent was an expert which tells the system how to act in the unstructured environment dynamics for specified actions. So, the reward function which provides the combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward was given below in the figure as the follows for the first trained agents.

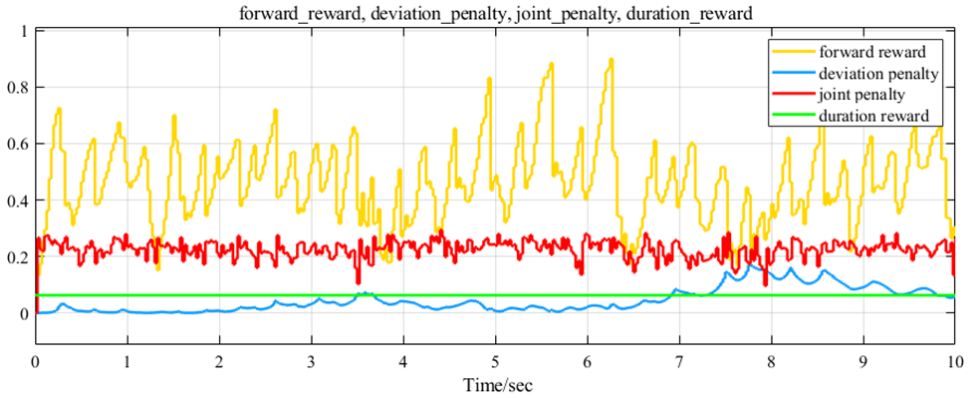


Figure 4.62: 3D-RL Induvial Reward for the first trained agent of the RL problem

As shown in the *figure 4.62* above there are four reward curves forward curves, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space. So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adaptive in the unknown environment dynamics and in order to responds the above figures for the possible state-action pairs of walking robot.

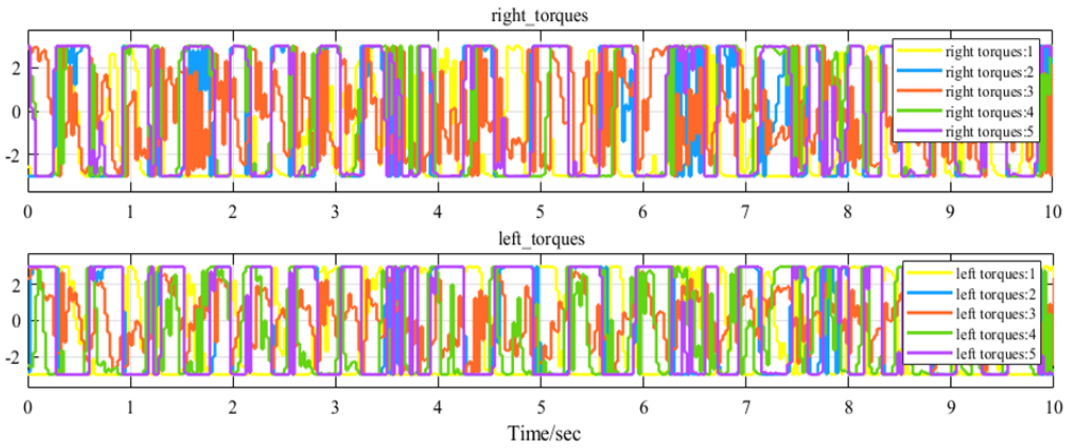


Figure 4.63: 3D-RL sequence of action for the first trial of trained agent

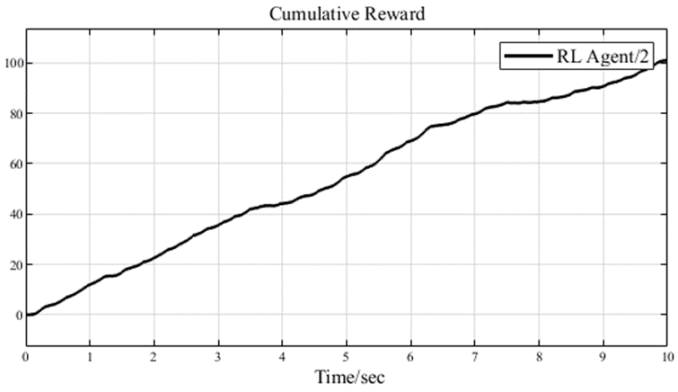


Figure 4.64: 3D-RL first trained agent for cumulative reward

To show how the first trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the figure below.

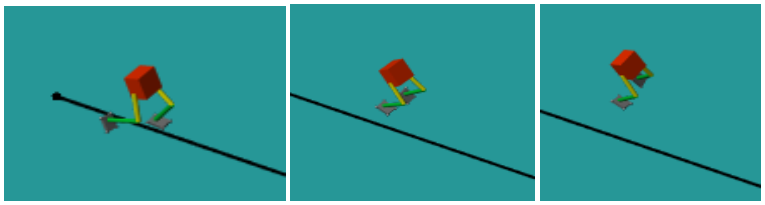


Figure 4.65: 3D walking robot the first training RL agent simulation result[a][b][c]

As I showed in the above figure the walking robot of the first trained agent based on the RL in the three-dimensional environment, the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. In the next section, I would like to show the second trained result for the biped walking humanoid robot in the 3D environment in the same manner as I had done in the above.

4.3.1.2.2 3D walking robot the second training RL agent

In the second trial to train the 3D walking biped humanoid robot with the RL agent, the required output for the action was well performed based on the more frequent training of the RL agent, such an agent as an expert which tells the system how to act in the unstructured environment dynamics for specified actions. So, the reward function which provides the combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward was given below in the figure as the follows for the second trained agents.

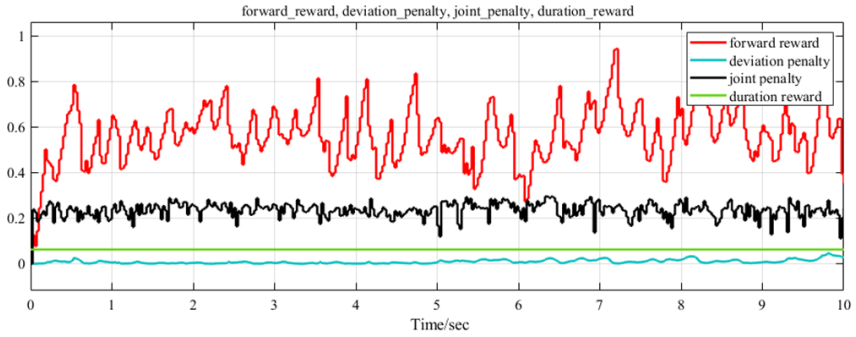


Figure 4.66: 3D-RL Induvial Reward for the secondly trained agent of the RL problem

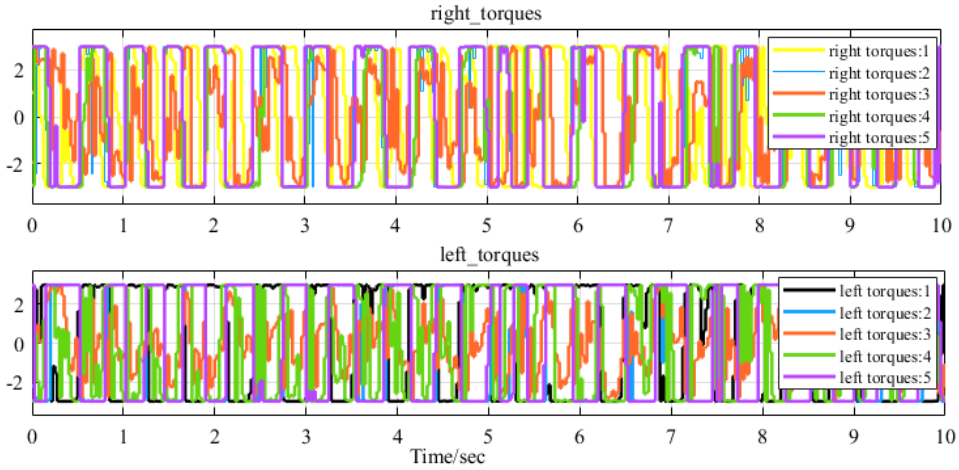


Figure 4.67: 3D-RL sequence of action for the secondly trial of trained agent

As we see in the figure 4.66 above there are four reward curves that showed in above, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space. So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust

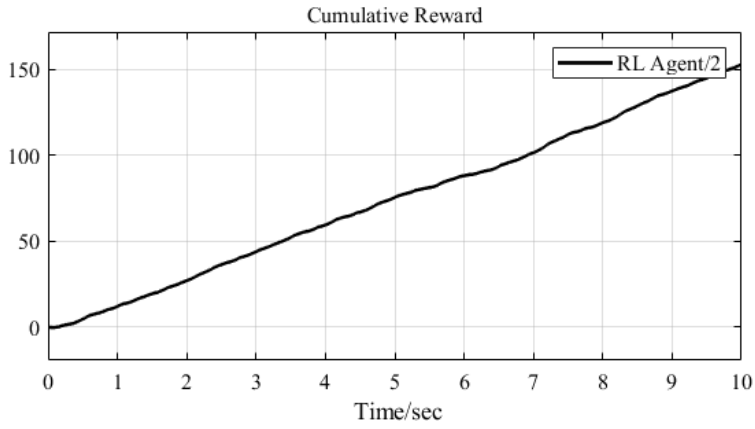


Figure 4.68: 3D-RL secondly trained agent for cumulative reward

and adaptive in the unknown environment dynamics and in order to responds the above figures for the possible state-action pairs of walking robot in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking biped for the specified torque command and this

reward can be assumed in the cumulative reward for minimize error this representation also mentioned as the figure shown above. To show how the second trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the secondly trained agent by providing the figure as shown in the figure below.

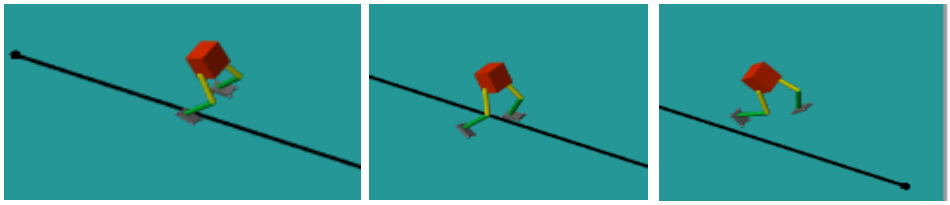


Figure 4.69: 3D walking robot the secondly trained RL agent simulation result[a][b][c]

As we see in the above figure the walking robot of the first trained agent based on the RL in the three-dimensional environment the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. In the next section, I would like to show the thirdly trained result for the biped walking humanoid robot in the 3D environment in the same manner as I had done in the above.

4.3.1.2.3 3D walking robot the third training RL agent

In the third trial to train the 3D walking biped humanoid robot, in the RL problem formulation, the required output of the action well performed based on the more frequent training of the RL agent, such an agent as an expert which tells the system, how to act in the unstructured environment dynamics for specified actions.

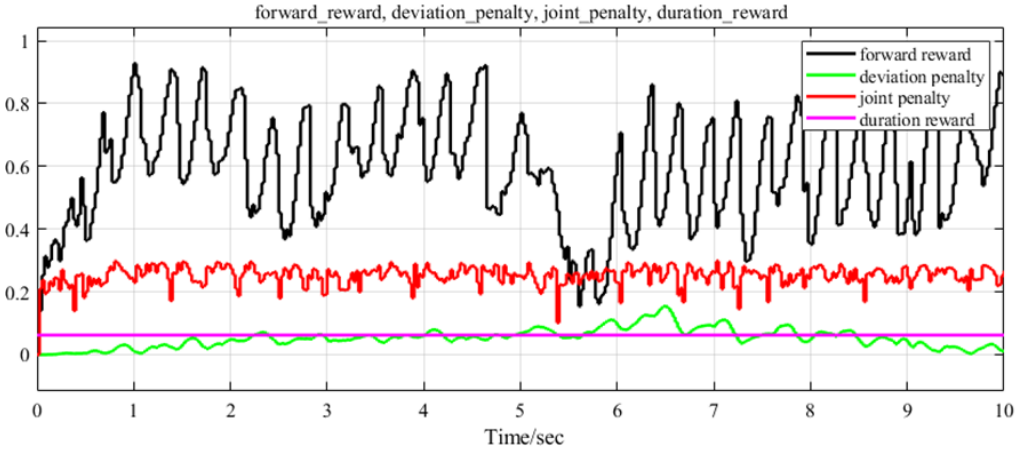


Figure 4.70: 3D-RL Induvial Reward for the thirdly trained agent of the RL problem

As we see in the figure above there are four reward curves forward curves, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space.

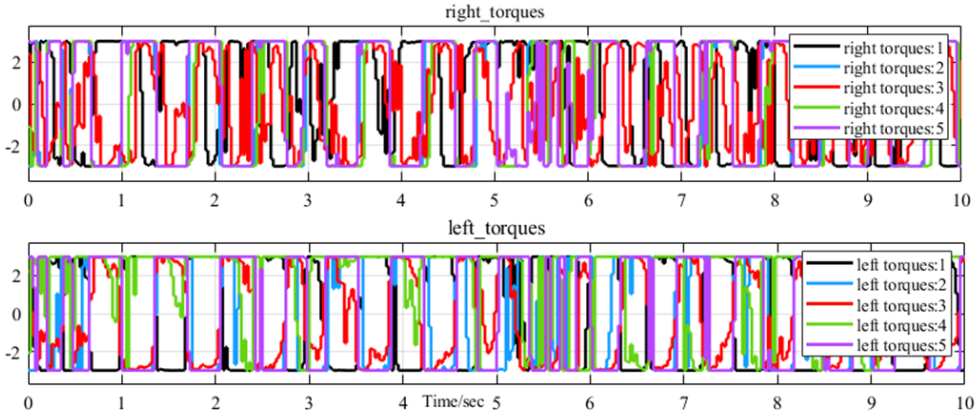


Figure 4.71: 3D-RL sequence of action for the thirdly trial of trained agent

So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adaptive in the unknown environment dynamics and in order to responds the above figures for the possible state-action pairs of walking robot in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking biped for the specified torque command and this reward can be assumed in the cumulative reward for minimize error this representation also mentioned as the figure shown below.

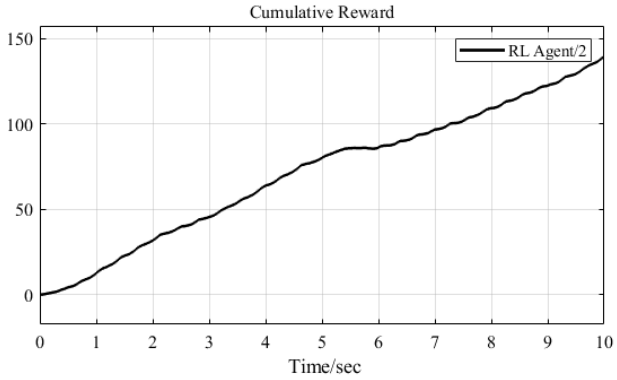


Figure 4.72: 3D-RL thirdly trained agent for cumulative reward

To show how the thirdly trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the figure below.

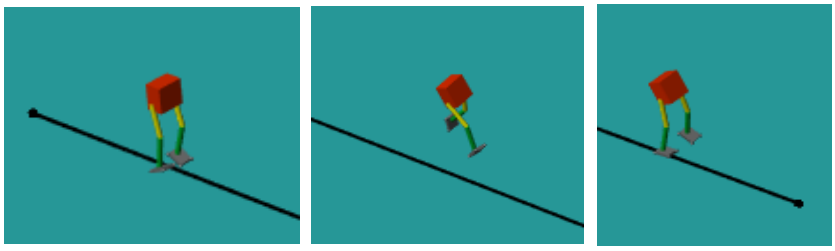


Figure 4.73: 3D walking robot the thirdly trained RL agent simulation result[a][b][c]

As we see in the above figure the walking robot of the first trained agent based on the RL in the three-dimensional environment the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. In the next section, I would like to show the fourth trained agent result for the biped walking humanoid robot in the 3D environment in the same manner as I had done in the above.

4.3.1.2.4 3D walking robot the fourth training RL agent

In the fourth trial to train the 3D walking biped humanoid robot, in the RL problem formulation, the required output of the action well performed based on the more frequent training of the RL agent, such an agent as an expert which tells to the system how to act in the unstructured environment dynamics for specified actions.

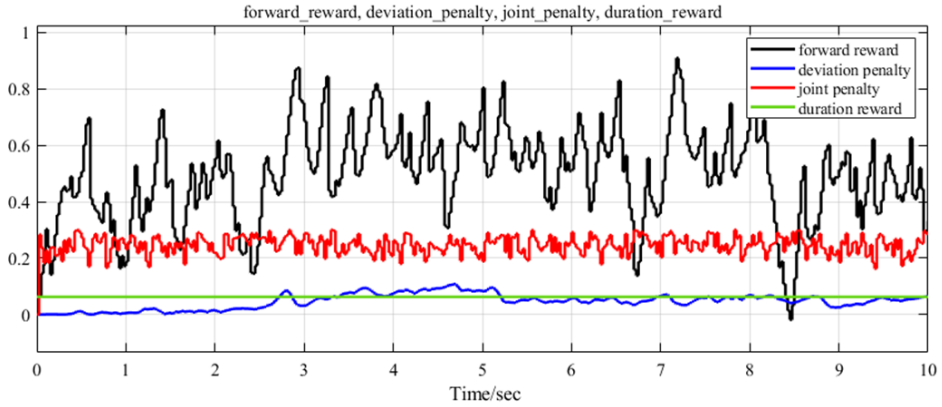


Figure 4.74: 3D-RL Induvial Reward for the fourth trained agent of the RL problem

As we see in the figure above there are four reward curves forward curves, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space.

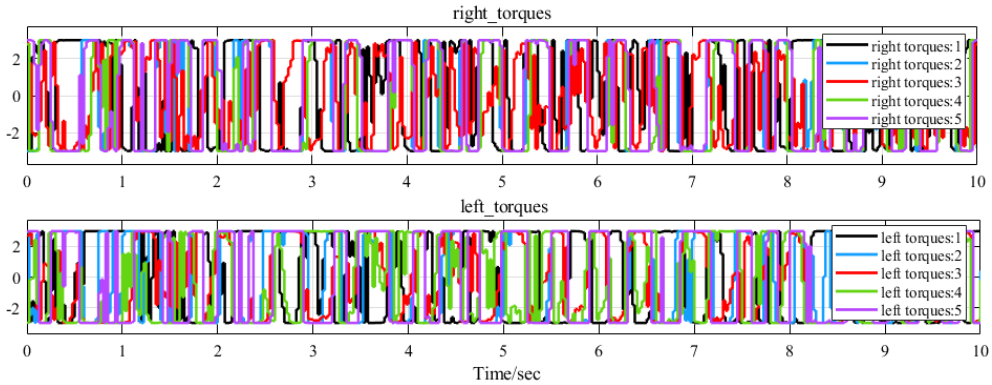


Figure 4.75: 3D-RL sequence of action for the fourth trial of trained agent

So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adaptive in the unknown environment dynamics and in order to

responds the above figures for the possible state-action pairs of walking robot in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking biped for the specified torque command and this reward can be assumed in the cumulative reward for minimize error this representation also mentioned as the figure shown below.

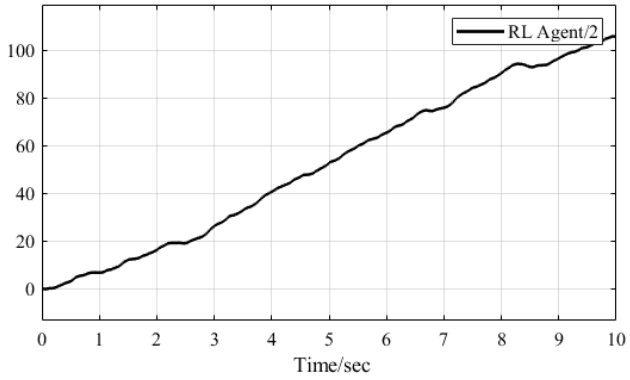


Figure 4.76: 3D-RL fourth trained agent for cumulative reward

To show how the fourthly trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the figure below.

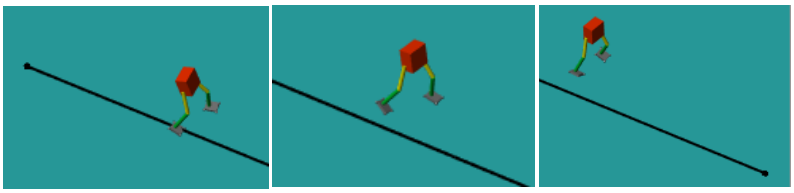


Figure 4.77: 3D walking robot the fourthly trained RL agent simulation result[a][b][c]

As we see in the above figure the walking robot of the first trained agent based on the RL in the three-dimensional environment the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. In the next section, I would like to show the fifth trained agent result for the biped walking humanoid robot in the 3D environment in the same manner as I had done in the above.

4.3.1.2.5 3D walking robot the fifth training RL agent

In the fifth trial to train the 3D walking biped humanoid robot, with the RL problem agent, the required output of the action well performed based on the more frequent training of the RL agent, such an agent as an expert which tells to the system, how to act in the unstructured environment dynamics for specified actions. So, the reward function which provides the

combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward was given below in the figure as the follows for the second trained agents.

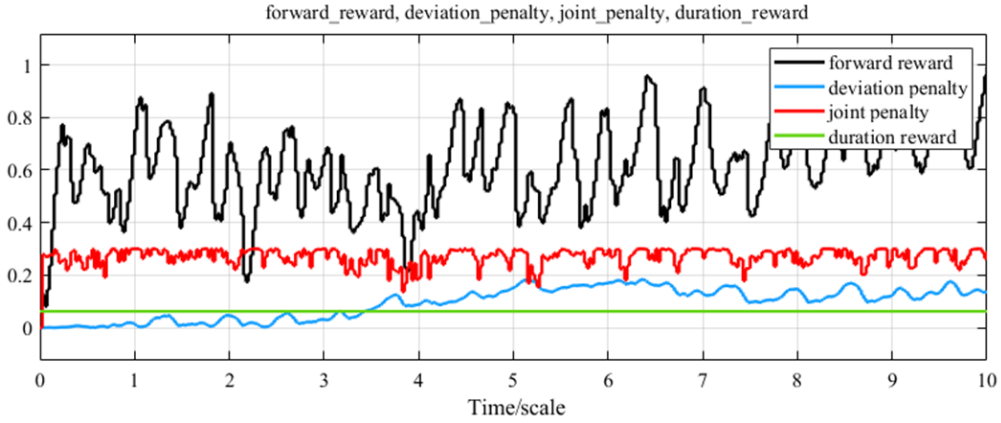


Figure 4.78: 3D-RL Induvial Reward for the fifths trained agent of the RL problem

As we see in the figure above there are four reward curves forward curves, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space.

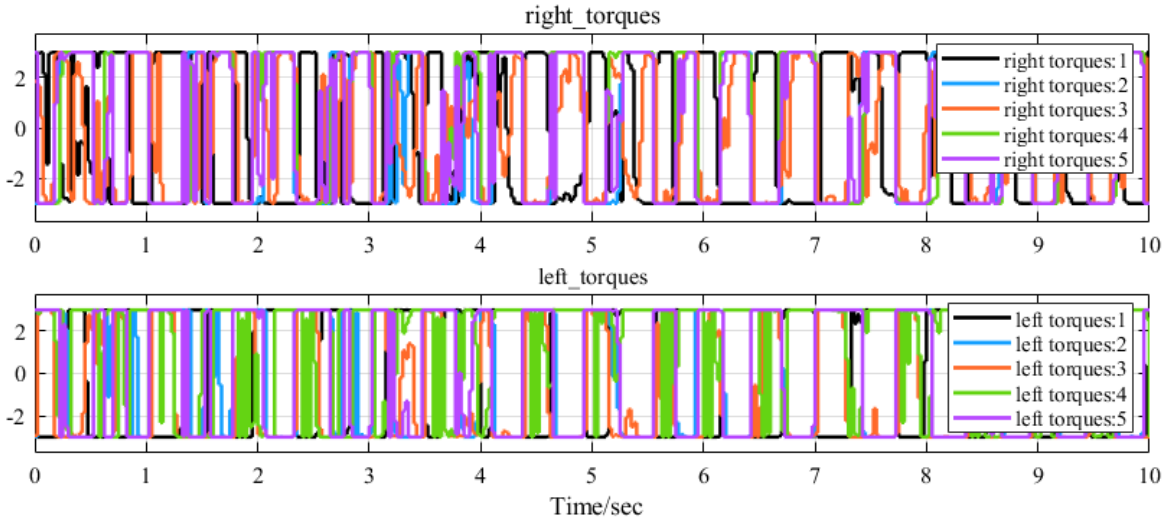


Figure 4.79: 3D-RL sequence of action for the fifths trial of trained agent

So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adaptive in the unknown environment dynamics and in order to responds the above figures for the possible state-action pairs of walking robot in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking biped for the specified torque

command and this reward can be assumed in the cumulative reward for minimize error this representation also mentioned as the figure shown below.

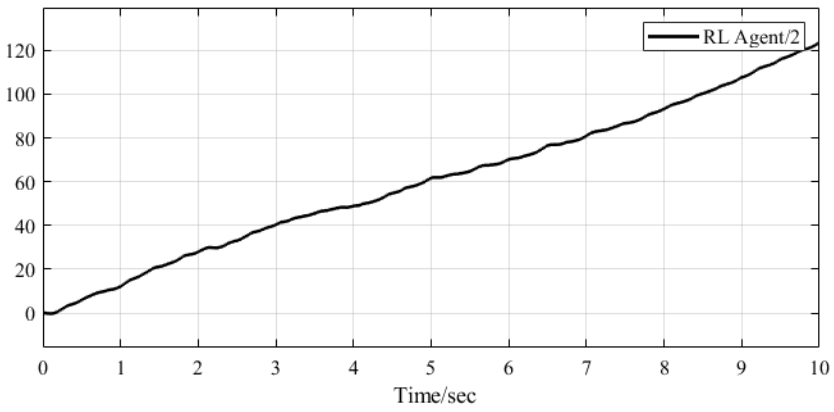


Figure 4.80: 3D-RL fifth trained agent for cumulative reward

To show how the fifth trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the first trained agent by providing the figure as shown in the figure below.

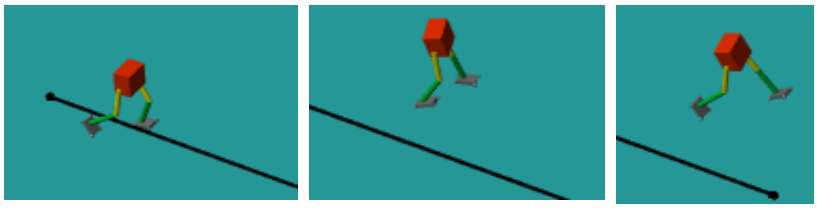


Figure 4.81: 3D walking robot the fourthly trained RL agent simulation result[a][b][c]

As we see in the above figure the walking robot of the fifth trained agent based on the RL in the three-dimensional environment, the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly. In the next section, I would like to show the sixth trained agent result for the biped walking humanoid robot in the 3D environment in the same manner as I had done in the above.

4.3.1.2.6 3D walking robot the sixth training RL agent

In the sixth trial to train the 3D walking biped humanoid robot, with the RL agent, the required output of the action well performed based on the more frequent training of the RL agent, such an agent as an expert which tells the system how to act in the unstructured environment dynamics for specified actions. So, the reward function which provides the combination of forward reward, deviation penalty, joint penalty and duration reward from the measurable observation of the environment along to the x, y and z by different gain of

forward reward scaling, deviation penalty scaling in the y and z and finally the torque penalty scaling provided by duration reward all the induvial reward was given below in the figure as the follows for the second trained agents.

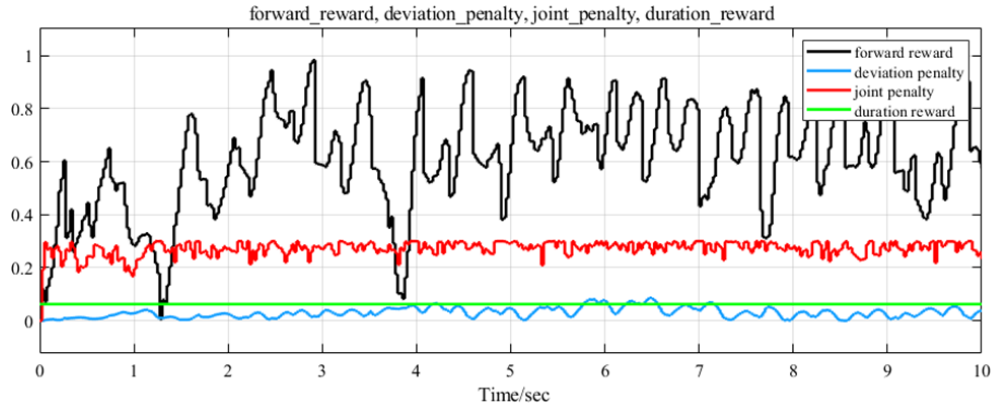


Figure 4.82: 3D-RL Induvial Reward for the sixth trained agent of the RL problem

As we see in the figure above there are four reward curves forward curves, the forward velocity curves, which allows the robots to walk in the given environments with the deviation penalty to joint penalty as well with the given time space.

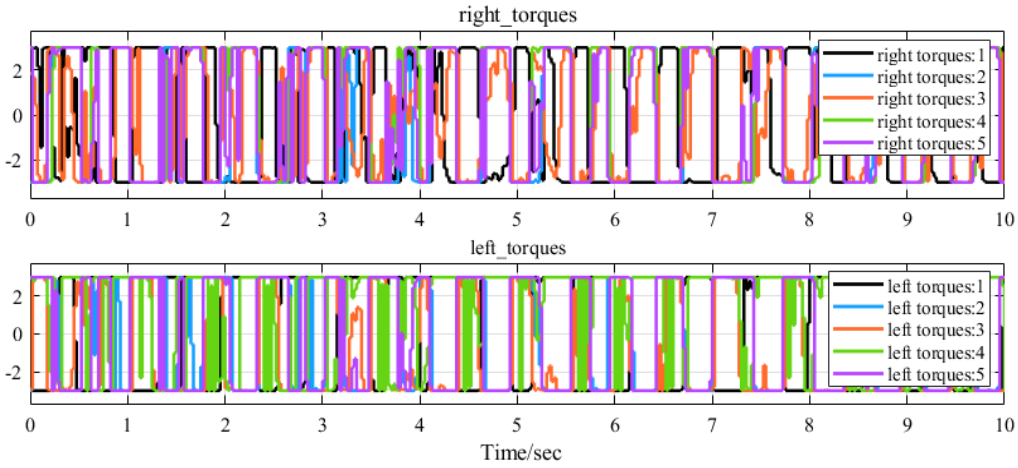


Figure 4.83: 3D-RL sequence of action for the sixth trial of trained agent

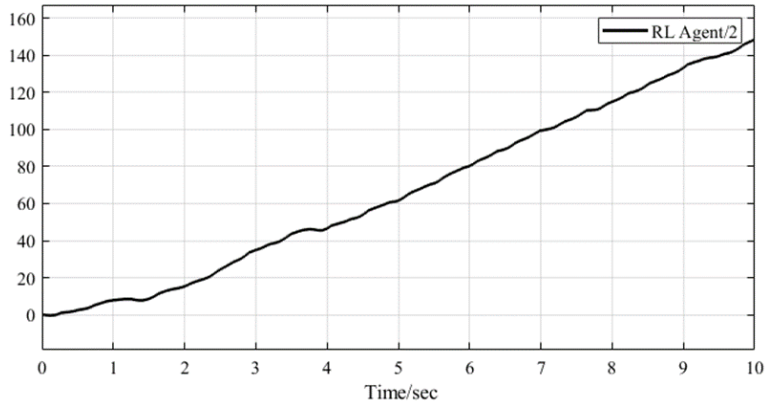


Figure 4.84: 3D-RL sixth trained agent for cumulative reward

So, the basic objective of the simulation in every trained agent was to made a walking robot to become more robust and adaptive in the unknown environment dynamics and in order to responds the above figures for the possible state-action pairs of walking robot in which the induvial reward of the forward velocity, deviation penalty, the joint penalty and the duration reward for the possible sequence of action in the walking biped for the specified torque command and this reward can be assumed in the cumulative reward for minimize error this representation also mentioned as the figure shown below. To show how the sixth trained agent behaves in the 3D walking robot, we can see the animation of the biped robot in which it follows the straight path in the unknown environment for the sixth trained agent by providing the figure as shown in the figure below.

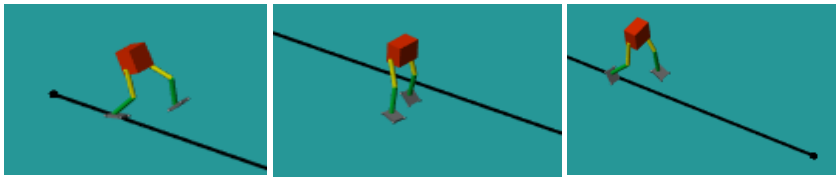


Figure 4.85: 3D walking robot the fourthly trained RL agent simulation result[a][b][c]

As we see in the above figure the walking robot of the first trained agent based on the RL in the three-dimensional environment the three figures marked [a] to [c] left to right shows how a robots walk in the given path from the start point to the goal point in the given environment dynamics accordingly.

4.4 Summary

4.4.1 Biped Locomotion and Balancing Strategies of Dinkinesh Humanoid

In the biped locomotion and balancing strategies for the Dinkinesh biped humanoid robot in this work, the walking biped humanoid robot was simulated in the MATLAB Simulink environment based on several method and controller based on linear inverted pendulum model with zero moment point (ZMP), center of mass stability measurement and model predictive control. Generally, the biped locomotion and balancing strategies of Dinkinesh humanoid based on the model predictive control compared with an existed work that reviewed in this paper that provided in the *table 4.1* below.

Table 4.1 Biped Locomotion and Balancing Strategies of Dinkinesh Humanoid based on the model predictive control compared with an existed work.

References	**Titles**	**Methodology**	**Gap**	**Improvements in this thesis**

(Mathew et al, 2017)	Model Predictive Control of Underactuated Bipedal Robotic Walking.	Rapidly exponentially stabilizing control lyapunov functions (RES-CLFs) and standard model predictive control (MPC) with feedback linearization.	They had used the dynamic modeling from the kinematics of Jacobian and most of the time the closed form of inverse kinematics problem was preferred in the design of humanoid robot, because of the kinematics singularity.	In this work, the closed form of inverse kinematics problem design was performed based on the arc tangent function for the inverse kinematics problems, which allowed the robot to be more coordinate stepping controllable, and it reduced the coordinate frame singularity which mathematically provided in the section [3.2.1.1].
(Marc et al, 2018).	*Model Predictive Control of Biped Walking on Humanoid Robot*	Feedback linearization along with computed torque.	They hadn't mentioned the stability of the gait in their simulation output.	The MPC controller in this work used the ZMP and COM position to allow the gait stability in the biped locomotion and balancing strategies. This stability was associated with the linear inverted pendulum model constraint of limiting with the walking parameters. The stability was evaluated and verified in software simulation in MATLAB Simulink in the animation.
(Scianca et al, 2020)	MPC for Humanoid Gait Generation: Stability and Feasibility	They had used as prediction model was a dynamically extended LIP with ZMP	In their paper the stability constraint was linked only with the future ZMP velocities to the existed state of the system model which just guaranteed the bounded CoM trajectory generation with respect to the ZMP trajectory	In this paper the stability constraint was linked both with the future ZMP velocities to the existed state of the system model which just guaranteed the bounded CoM trajectory generation with respect to the ZMP trajectory and Center of mass of the robot. The stability was evaluated and verified in software simulation in MATLAB Simulink in the animation.

(Hong et al, 2020) al	Real-Time Constrained Nonlinear Model Predictive Control on SO(3) for Dynamic Legged Locomotion	analytic Jacobians with gradient and the Gauss-Newton Hessian approximation of the objective function	The analytical Jacobian most of the time they are not suitable for the motion kinematics formulation in the complex humanoid robot thus because of the singularity and coordinate frame alignment.	In this paper the analytical joint solution was computed with the closed form of the inverse kinematics problem, this closed form inverse kinematics problem was just improved the joint space alignment of the walking robot in the two dimensional and the three-dimension coordinate frames.
(Wieber, 2006)	*Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations*	Linear MPC with ZMP Preview Control scheme	They had used indirect position control from the torque perturbation and it is not suitable for the real-time control of the RC servo motor and it needs the extra tachometer sensor reading.	In this paper, I had used the direct position control was used in the concept of zero moment point and the center of mass projection within the base frame and global frame of the robot the base frame was the foot frame and the global frame also was the frame attached in the waist of the robot.

4.4.2 Biped Path Planning Strategies of Dinkinesh Humanoid Robot

In this work, the path planning strategy was used with zero moment point trajectory generation based on model predictive control of the center of mass trajectory generation based on different paths of waypoints, and generally the path planning method was used with model predictive control and the measurement assigned to tracking the given trajectory was based on the different path like curricular, square and star path of waypoints in this waypoint the path planning method of model predictive control used to make the robot to walk from the input of the reference of the two dimensional x, y and sometimes r direction was assigned to made a robot to walk into the given trajectory. Generally, in this works the walking path of the robot was verified in the simulations based on the MATLAB Simulink simMechanics library which was explained in the discussion of the above section.

4.4.3 Integrating robotics and Artificial intelligence to Dinkinesh Humanoid robot.

In this work the integration of robotics and artificial intelligence to biped humanoid robot Dinkinesh, the deep reinforcement learning with a deep deterministic policy gradient for the

continuous state-action for the walking of biped humanoid robot was used for the walking control of the robot without system modelling of the walking plant, the results were verified based on the analytical with simulation in the animation in the MATLAB Simulink with simMechanics MATLAB libraries. The state of art of comparison of the results of the study was compared with the others works according to the literature reviewed in the related works accordingly, in this works at the better results was obtained in which when it was compared with the other works of the same or the method of application of deep reinforcement learning with deep deterministic policy gradient of an agent to train or teach the neural networks with the deep deterministic policy gradient and the state of art of comparisons was provided in *the table 4.2 below.*

Table 4.2 The state of art comparison for the integration of robotics and artificial intelligence from the existed literature reviewed.

References	Titles	Methodology	Gap	Improvements
(Jun. M et al, 2004).	A Simple Reinforcement Learning Algorithm For Biped Walking.	Model-based RL algorithm for biped walking robot in which the robot learned appropriately for placing the swing leg.	One of the gaps of their study was the RL problem representation was computed with discrete state space action within a derived model dynamic this was not the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a mathematical interpretation of the blackbox.	This work improves the walking robot reinforcement learning problem representation by representing continuous state action without a derived model dynamic this was the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a mathematical interpretation of the blackbox. This leads the application of DRL of DDPG to become the general artificial intelligence.
(Erik et al, 2010)	The Design of LEO: a 2D Bipedal Walking Robot for	Discreate state action hierarchical Reinforcement Learning.	The agent they had used in their paper was not continuous state instead they had used the discrete	In this work the agent called DDPG for the continuous state-action had used in the

	Online Autonomous Reinforcement Learning.		markov model process [MDP]. The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence.	continuous markov Decision process [MDP]. The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence.
(Moham mad et al, 2021)	Robust biped locomotion using deep reinforcement learning on top of an analytical control approach	Based on genetic algorithm (GA) and proximal policy optimization (PPO)	The agent they had used in their paper was not continuous state instead they had used the discrete markov model process [MDP] The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence. The network they had used was only actor which leads to the learning agent to less autonomous.	In this work the agent called DDPG for the continuous state-action had used in the continuous markov Decision process [MDP]. The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence The other improvements in this works were the network that used was not only actor which leads to the learning agent to less autonomous, but also, the critic network was there thus a network just allowed less computational burden.
(Diego .R & Sven.B, 2021)	DeepWalk: Omnidirectional Bipedal Gait by Deep Reinforcement Learning	Innovative deep reinforcement learning.	The locomotion behaviors are limited to accomplished by a single control policy	In this works the locomotion behaviours was not limited to accomplished by a single control policy.

CHAPTER FIVE

HARDWARE DESIGN DISCUSSION AND RESULTS

5.1 Hardware Design

In this chapter, I had discussed the hardware system design, implementation and results of Dinkinesh biped humanoid robot, the hardware system of the biped humanoid robot was designed with Raspberry pi 4B 2GB graphical processing unit which allows to compute AI system within it and AVR ATmega2560 microprocessor integrated with microcontrollers for the real time motion control of the biped walking humanoid robot for the waking of biped mechanisms, all the design was conducted with scientifically interpreted with relevant coding and debugging with standard C/C++ and python language.

The mechanical design, programming control, electronics soldering and PCB design was basically implemented for in order to develop the Dinkinesh biped humanoid robot, however the first step was the mechanical design, then next electronics assembling and finally testing and implementation was done accordingly and such a constructed algorithm can be described in the section below independently.

5.1.1 Mechanical Design and Assembly of the Dinkinesh Biped

The biped humanoid robot in this thesis that I called Dinkinesh was mechanically assembled with 20 joint actuators and the total links of 11 links and I had developed the biped humanoid robot called Dinkinesh from the scratch and all the materials I had used for the hardware design was made from the white light metals which extracted from the waste florescent tube and I was found it very light weight metal that can be applicable to the robotics prototyping application, lastly I want to describe the each parts of Dinkinesh biped humanoid mechanical assembly and it's process below accordingly.

5.1.1.1 The Leg part Design and Assembly

The first design of the biped humanoid robot Dinkinesh was started from the mechanical assembly and design of the two parts with its own frame assignment of the walking legs dynamics from the scratch. The hardware design of the mechanical system of Dinkinesh biped humanoid robot contained of two legs which designed from the scratch, the metal parts were hacked from the through of the florescent lamp holders and the servo motor with M3 bolt and nut for the tithing of the joints in the standard configuration formats. Such system designed with each of the legs presented in each step in the section given below accordingly.

Figure 5.1: Dinkinesh Biped humanoid hardware right leg configuration and assembly.

As shown in the figure above the right leg mechanical assembly were illustrated in the figure 5.1 in above with six servo motors in the ankle section the roll and pitch motion described in the two motors of 180 degree and one motor in the knee section for the knee pitch motion and the three servo motors for the hip pitch, roll and yaw motion accordingly, this was allowing the legs to be dexterous motion flexibility to the three-dimensional coordinate work space which most likely allow the robot to attain humanlike walking and appearances.

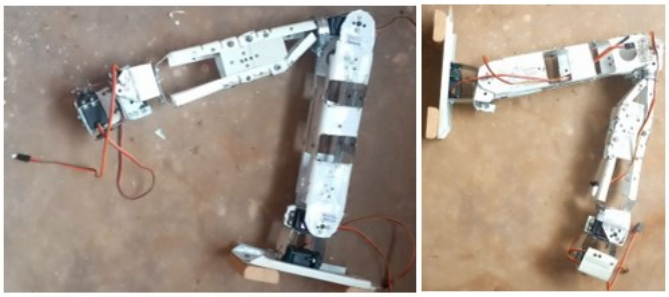


Figure 5.2: Dinkinesh Biped humanoid hardware right Leg Configuration in the knee bending backward

In the similar manner in the figure 5.3 as shown in the figure below the left leg mechanical assembly were illustrated in the figure 5.3 below with six servo motors in the ankle section the left roll and pitch motion described in the two motors of 180 degree and one motor in the knee section for the left knee pitch motion and the three servo motors for the left hip pitch, roll and yaw motion accordingly, this was allowing the legs to be dexterous motion flexibility to the three-dimensional coordinate work space.

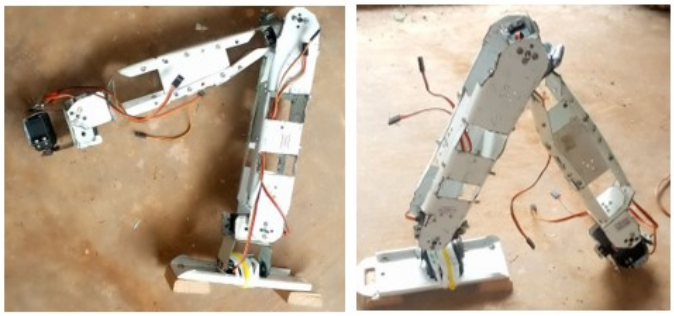


Figure 5.3: Dinkinesh biped humanoid hardware left leg assembly and configuration.

So, generally the hardware design of Dinkinesh biped humanoid robots within the legs parts were assembled in the smart ways accordinglly described in the privious steps of the design section and descrption in the above and the overall both legs parts described in the figure below as the end.

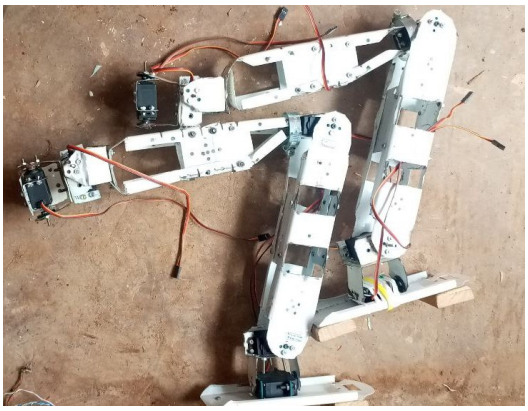


Figure 5.4: Dinkinesh biped humanoid hardware both right and left leg assembly and configuration.

5.1.1.2 The hand part design and assembly

In the case of the hand parts design of Dinkinesh biped humanoid, the same types of motor and metal casing was used in the difference of the degree of freedom of the hand's configuration and assembly process. Both of the left and right hands of the Dinkinesh biped humanoid contained of three servo motors to emulate the hands of humanoid characteristics, such an integrated system design of the hand for Dinkinesh biped humanoid was designed in the flexible manner with lower operating coast were described in the section below.

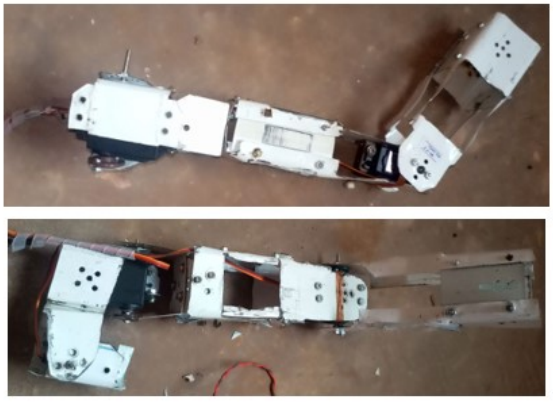


Figure 5.5: Dinkinesh biped humanoid hardware left hand assembly and configuration.

In similar ways the right hands of the Dinkinesh biped were illustrated in the figure 5.6 which contained of shoulder roll and pitch as well as the elbow pitch for the pitch motion and it only allowed to the three directions of motion for these thesis regard.

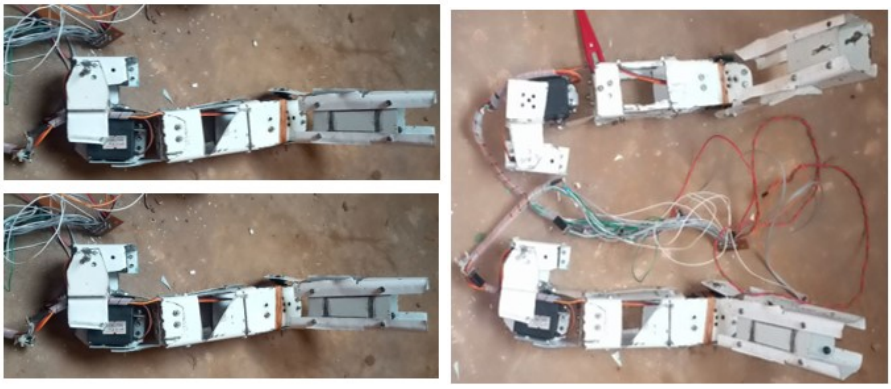


Figure 5.6: Dinkinesh biped humanoid hardware right hand assembly and configuration.

And finally, the mechanical parts of the hardware design of the Dinkinesh biped humanoid robot hands were described in the figure 5.7 which contained of the entire system which could allow the biped humanoid robots to be full humanoid robot. So actually, the hand parts also contained of the 6DOF such that it could capable of six different versatile motion of the hands around the kinematics workspaces.

Generally, the hand parts mechanical system design of the biped was shown that as it was so nice and plentifully achieved in successes. In the next section the upper torso part design and assembly were discussed accordingly.

5.1.1.3 The upper torso part design and assembly

The upper torso part design and assembly of Dinkinesh biped humanoid robot design consideration was implemented and the two hands and the torso design were considered, in this case around 6 servo motors was used to build the upper body of the Dinkinesh biped humanoid robot, this was considered and the flexible motion control and manipulation was implemented and tested.

5.1.1.4 The Overall Full humanoid of Dinkinesh biped humanoid robot design and assembly

In the previous sections, I had discussed the mechanical design of Dinkinesh biped humanoid robot independently like the two legs assembly, upper torso part design and assembly, the head part design and assembly, now in this section we had going to see the overall full humanoid of biped robot, with like lower hip design of the biped humanoid robot design was considered and the other thing out there was, the upper body design of the biped humanoid robot was considered.

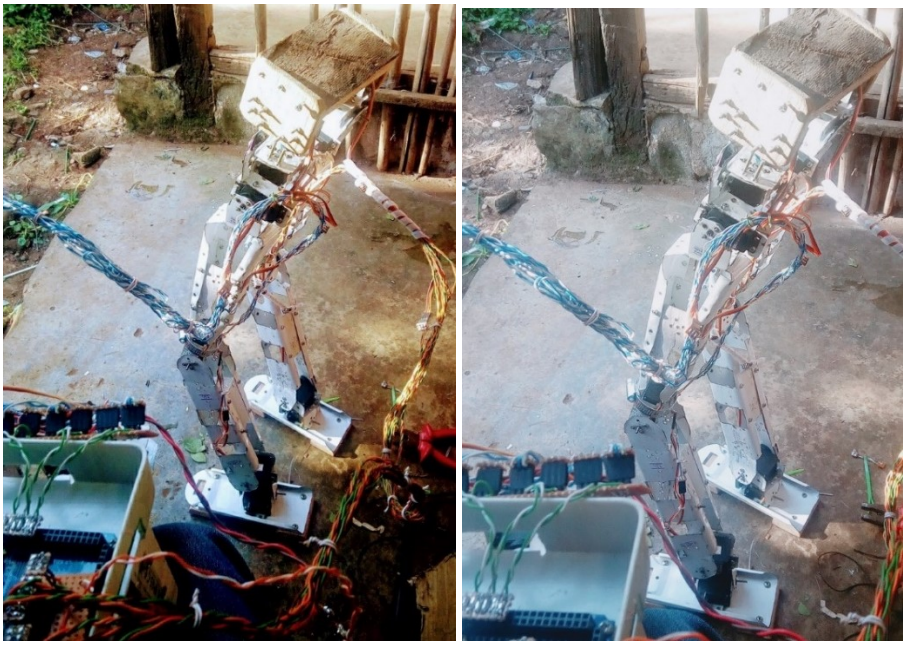


Figure 5.8: The lower hip mechanical part design of Dinkinesh biped humanoid robot

So, actually the 12 DOF robotics leg design for the Dinkinesh biped humanoid robot was successfully implemented for the real time implementation for the balanced postural stability of the biped walker humanoid robot.

5.1.2 The Electronics prototyping and PCB design assembly

The electronics parts for the assembly of the robot contained a simple electronics arrangement for the processors with some peripherals like servo motors, power conversion an amplification outlet.

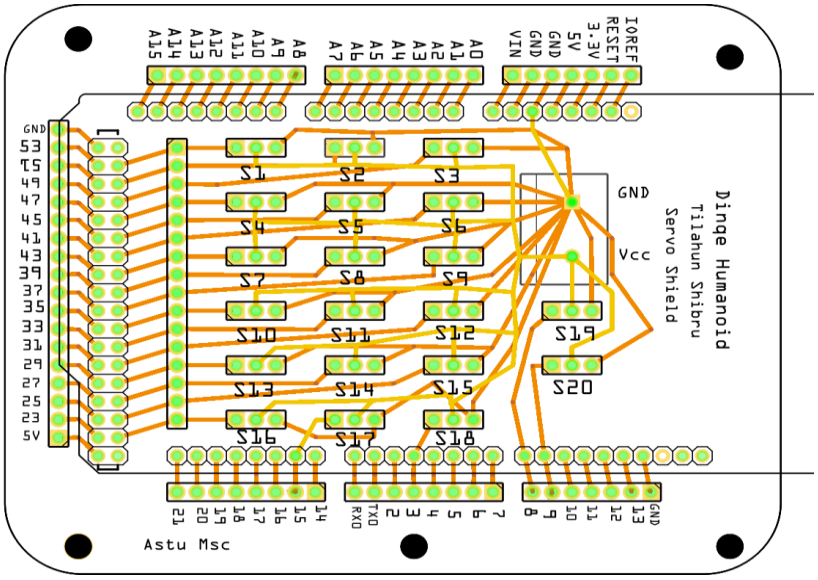


Figure 5.9: Electronics parts and PCB assembly design of Dinkinesh biped humanoid robot

So actually, the power supply design of Dinkinesh biped humanoid robot should contained of some useful electronics arrangements and to give an important regulated DC output power of 6 volt and an approximated output power of 2.4 ampere, the typical power supply design to drive the robotics system that's includes of 7805 voltage regulators, some peripherals and power MOSFET for the further power amplification for the output.

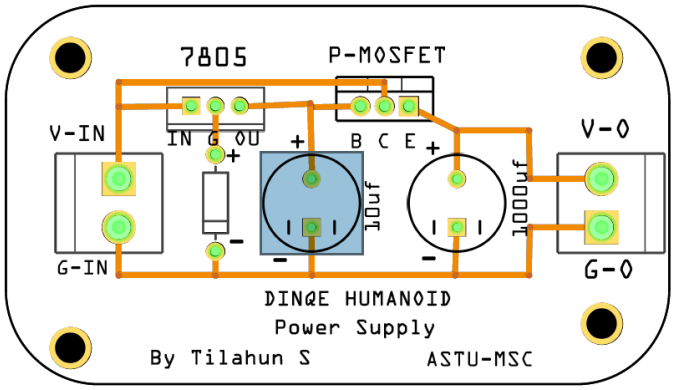


Figure 5.10: Power Supply Assembly design of Dinkinesh biped humanoid robot

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

6.1.1 Simulation Design and Implementation

In this thesis paper, the simulation design and implementation of Dinkinesh biped humanoid robot was presented based on the robot that could able to have the bipedal locomotion and balancing strategies, the ability to plan the path trajectory and walk through the planned trajectory and finally Dinkinesh biped humanoid robot powered with AI algorithm in order to teach the robot to walk without modeled system dynamics at the end the robot walked efficiently and the respective features of the robot with simulation environment was implemented and tested in MATLAB Simulink environment successfully thus, accordingly the conclusion and recommendation provided in the section below.

6.1.1.1 Biped Locomotion and Balancing Strategies of Dinkinesh Humanoid

I introduced all the software platforms that was used in this study. The Dinkinesh in the software environment is the biped humanoid robot that designed in the Matlab2019 that used to implement and test all of my work in the software and most of parts was implemented and tested in the simulation environment. Finally, I also presented all the available insights to program the Dinkinesh as well as the permissible programming languages.

I had introduced and defined several terms that are related to the locomotion stability. The concepts include of the Center of Mass (CoM), Center of Pressure (CoP), Zero Moment Point (ZMP) and support polygon. I reviewed the primary locomotion classifiers, such as model-based, model-free, passive and active walker. After that, I presented the most popular stability criterion for humanoid robots which is the ZMP criterion. I also discussed three physical models that can be used to simplify the robot's dynamics, which are the Cart-Table, Linear Inverted Pendulum (LIPM), and the Inverted Pendulum (IPM). A typical ZMP and COM based walking engine consists of several primary modules with the help of Model Predictive Controller (MPC). I presented each of the modules in details while explaining its working mechanism alongside its inputs and outputs. I address the process of integrating these modules together to produce an efficient walking engine able to make the biped walk with the desired gait. Developing a quality, fast and reliable locomotion is a complicated

task that is essential for adopting humanoid robots into the human's environments. However, without an excellent balancing ability, the humanoid will not be as useful as we desire.

The walking pattern generation so as to become an efficient, the stability measurements was considered with a relevant mathematical optimization with application of zero moment point and the center of mass of the robot, which allow the robot to become stable at standing and while in walking. The inverse kinematics problem based on the closed form in the frame of robot coordinate plays the other improvement in this work, that improve the joint singularity that reduced the robot DOF by the number of the singularity, thus the closed form of inverse kinematics problem thus just improved the relation of the foot frame to the waist frame to become more consistent.

In comparison with Mathew et al, they had used the dynamic model from the inverse kinematics of Jacobian this resulted a robot frame singularity and most of the time the closed form of inverse kinematics problem was preferred in the design of humanoid robot, because of the solution for kinematics singularity. To solve this problem, in this work, the closed form of inverse kinematics problem design was performed based on the arc tangent function for the coordinate frame of the robot joint, which allowed the robot to become more coordinate stepping controllable, and it reduced the coordinate frame singularity which mathematically provided in the section [3.2.1.1].

In the same manner in comparision to Hong et al, the analytic Jacobians with gradient and the Gauss-Newton Hessian approximation of the objective function, the analytical Jacobian was used and it was not suitable for the motion kinematics formulation in the complex humanoid robot thus because of the singularity and coordinate frame alignment. In this work, the analytical joint solution was computed with closed form of the inverse kinematics problem, this closed form inverse kinematics problem was just improved the joint space alignment of the walking robot in the two dimensional and the three-dimension coordinate frames.

In comparision to work of Scianca et al , MPC for humanoid gait generation: stability and feasibility, they had used a prediction model that was dynamically extended LIP with ZMP, however in their paper the stability constraint was linked only with the future ZMP velocities to the existed state of the system model which just guaranteed the bounded CoM trajectory generation with respect to the ZMP trajectory, in this work the stability constraint was linked both with the future ZMP velocities to the existed state of the system model which just

guaranteed the bounded CoM trajectory generation with respect to the ZMP trajectory and center of mass of the robot. The stability was evaluated and verified in software simulation in MATLAB Simulink in the animation.

Generally, the biped locomotion and balancing strategies for Dinkinesh biped humanoid robot was successfully implemented in the Matlab Simulink environment with the help of simMechanics animation in the mechanics library, the fundamental background to design the biped humanoid robot, that allowed the walking to become smoothly and efficiently, so for this concern the stability measurement played the first place and the physical system representation of linear inverted pendulum as walking constraint was leads to the successful results. So, generally the MPC controller for the walking plant with the physical model of the double linear inverted pendulum with the aid of zero moment point and center of mass of the robot played other successful walking in the trajectory generation and with other optimization method of the walking biped humanoid robot.

6.1.1.2 Biped Path Planning Strategies of Dinkinesh Humanoid Robot

In, the case of path planning strategy, the model predictive control was used, based on the three different types of complicated path of the robot that includes of circular, square and star path trajectory was implemented with the application of zero moment point trajectory generation based on model predictive control of the center of mass trajectory generation. As the result, the biped humanoid robot that walked in the smooth terrain that was animated in Matlab Simulink environment and it gives a good result and the robot was just reached to the goal point based on the given time frame accordingly to the velocity tracking this just required to test the robot to reach to the target from the start point to the goal point successfully.

The biped humanoid robot that was designed, in this works just plan the path based on different paths of waypoints, and generally the path planning method was used with model predictive control and the measurement that assigned to the tracking for the given trajectory was obtained and based on the different path of waypoints, that includes of the curricular, square and star path of waypoints, in this waypoint the path planning method of model predictive control also used to test the robot to walk from the input reference of the two-dimensional x, y and sometimes r direction was assigned to made a robot to walk into the given trajectory in the three dimensional plane.

Generally, in this works the walking path of the robot was verified in the simulations environment based on the MATLAB Simulink environments with helps of simMechanics library that was explained in the discussion and result part in the above section and all the expected path waypoints was measured based on the required time frames.

6.1.1.3 Integrating robotics and Artificial intelligence to Dinkinesh Humanoid robot.

In the case of the integration of robotics and artificial intelligence method to biped humanoid robot (Dinkinesh) in the MATLAB Simulink environment was implemented and tested successfully, for these application the general AI was considered, that called the deep reinforcement learning with a deep deterministic policy gradient for the continuous state action for the walking of biped humanoid robot, that was used for the walking control of the robot without further mathematical modelling of the walking plant, even without the complex mathematical derivation and the resulting output in the simulation environment resulted in the successful implementation, that was verified based on the analytical with simulation with the help of animation in the MATLAB Simulink environment with simMechanics of MATLAB libraries.

The state of art of comparison of the results of this works and other papers according to the literature that reviewed in the related previous works accordingly and final I draw a conclusion accordingly, in this works at the better results was obtained when it was compared with the other works of the same or different method of application of deep reinforcement learning with deep deterministic policy gradient of an agent to train or teach the neural networks with the deep deterministic policy gradient and the state of art of comparisons was provided below. In the comparision with jun M et al, in there paper of a simple reinforcement learning algorithm for the biped walking humanoid, they had used the model-based reinforcement learning algorithm for biped robot, in which the robot learned appropriately for placing the swing leg in the walking. However, they had represented the reinforcement learning that was computed with discrete state-action within a derived model dynamic this was not the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a mathematical interpretation of the blackbox. This work improved the gaps of above work in the biped walking robot with the reinforcement learning problem by representing continuous state-action, without a derived model dynamic, this was the fundamental requirement of the deep reinforcement learning over an advanced control theory which was supposed to be based on the unmodelled system dynamics of the plant without a

mathematical interpretation of the blackbox. This leads the application of deep reinforcement learning of deep deterministic policy gradient to become the general artificial intelligence.

Similarilly in the works of Erik et al, in the design of LEO: a 2D bipedal walking robot for online autonomous reinforcement learning. They had used the discreate state-action hierarchical reinforcement learning, the agent they had used in their paper was not continuous state instead they had used the discrete markov model process [MDP]. The agent that trained with continuous state action pair were preferable to BWR that leads to promise of AI toward general artificial intelligence. In this work the agent called DDPG for the continuous state-action had used in the continuous Markov Decision Process [MDP].

In the other hand Mohammad et al, in there works of robust biped locomotion using deep reinforcement learning on top of an analytical control approach. They also designed RL for the biped humanoid robot based on genetic algorithm (GA) and proximal policy optimization (PPO). In their works they had used discrete Markov Model Process [MMP]. However, in this works, the agent that trained with continuous state-action pair were preferred to BWR that leads to promise of AI toward general artificial intelligence and in their works the network they had used was only actor which leads to the learning agent to less autonomous. In this work the agent called DDPG for the continuous state-action had used in the continuous Markov Decision process [MDP].

The agent that trained with continuous state-action pair were preferable to BWR, that leads to promise of AI toward general artificial intelligence. The other improvements in this works were the network that used was not only actor that leads to the learning agent to less autonomous, but also, the critic network was there thus a network just allowed less computational burden and also it reduce the complex mathematical derivation. In the comparision with works of Diego .R & Sven.B, DeepWalk: in the omnidirectional bipedal gait by deep reinforcement learning, innovative deep reinforcement learning, the locomotion behaviors also limited to accomplished by a single control policy, in this works the locomotion behaviours was not limited to a single control policy. The reinforcement learning can solve the complicated control problems with the simple representation this complicated problem can used a neural network that could capable of handling continuous or the high dimensional state and action space. The optimal and an updated state output for this complicated control process depending on the chosen of the good state, actions, and reward function was extremely important, thus by assuming that the reinforcement learning requires

a lot of simulation data thus the DDPG could contained of high variance algorithmic and that just run several times and validate results.

6.1.2 Hardware System Design and Implementation

In the hardware design and implementation for the biped humanoid robot, I introduced all the hardware platforms that were used in this study. The Dinkinesh also the biped humanoid robot that was designed from the scratch that used to implement and test some of my work in the hardware and most of parts was implemented and tested in the software simulation environment.

6.1.2.1 Biped Locomotion and Balancing Strategies of Dinkinesh Humanoid

In this work the design of Dinkinesh biped humanoid in the hardware environment was modeled and thus some of the work for the mechanical motion of the walking and other motion, the hand motion and the head were tested and implemented in the hardware environment. The application and selection of the motor, the mechanical parts and lightweight components, the power supply selection, the wiring and soldiering was the main factors for the successful design and implementation of the biped humanoid robots in the hardware and prototypes in addition to this the selection of processers and sensors plays the other most important consideration. In this works the construction and development of the biped humanoid robots contained of some sensors, lightweight metals and some useful motors with it and thus Dinkinesh biped humanoid robot were described by the 20DOF and mechanical parts and some of the useful software that control it inside the processer or the computer board. Generally, the implementation and design of the biped humanoid robot, was successful from the view point of the financial funding capability, the knowledge and skills and finally the operating time for conducting the research, this because of the development of the humanoid robot research is not the works that were conducted to be fully operational as the induvial level and in one discipline alone, thus the achievement of this work mainly was the hardware design and implementation of the biped humanoid robot Dinkinesh.

6.2 Recommendations

Generally, the works of the study that's mainly relied on the locomotion and balancing strategies, path planning and the integration of AI to the biped humanoid robot, the positive features and the feature improvements and the difficulties that associated with the field of the study was recommended in the section below accordingly. The stable humanoid locomotion is not a closed problem, this work has proposed and validated a feasible method

for generating stable walking patterns. The conclusions of the presented work are detailed as follows:

1. **Trends of humanoid robots' construction**
    - Humanoid robot progress to bipedalism is optimal possibility for developing of an intelligent machine. Those ways, some progress on walking robotics was explained with the emphasis on the results in the few research group around the whole world and also the walking robotics is an important and attractive research areas in the robotics, not enough applications have been made yet. For the future of the walking robotics design was considered in such ways of how to improve those situations around it.

2. **About human locomotion and humanoid robot locomotion**
    - The normal bipedal gait is achieved with a complex combination of automatic and volitional posture components. The normal walking requires the stability to provide antigravity support of body weight, mobility of the body segments and motor control to sequence multiple segments while transferring body weight from one limb to another.
    - The stability criteria for obtaining stable biped walking have been outlined, especially the "Zero Moment Point (ZMP)" criterion, which is the most popular and most often applied to humanoid robot in the world.
    - And finally, the further research related to humanoid-human cooperation should be made in order to handle ordinary tasks in the human environments.
3. **About Path Planning and Obstacle Detection**
    - In the case of the path planning and obstacle detection for the robotics environment plays the other most important contribution for the autonomous navigation for the robotics system, especially for the biped humanoid robot to reach to the required goal points to operate for the given tasks, however in this works, the path planning method with the model predictive control with the help of zero moment point was considered, yet this is not enough for the robot to reach into the complicated environments such as to consider even in the unstructured environments.
    - So, actually the robot should be capable of handling the obstacle so as to detect the obstacle and handling before reach to the goal points they should have a search method, this will be the future works of the study by adding the search method

into the robotics system through the usage of an accelerated a high computing powerful device.

- The autonomous functionality of the humanoid robot also mainly depends on an efficient path planning and obstacle detection system, however in these works the scope also limited only in the path planning system, however there is a better way to improve the autonomies by using the classifiers like A*, D* and rapidly random tree and etc.

4. About the Integration of AI and Robotics

- In this works the integration of robotics and artificial intelligence method in the biped humanoid robot Dinkinesh for the area of trained agents in the walking tasks of application, however the AI discipline can be applicable to almost in any parts of the design of humanoid robot development, like the vision and brain net, bio-chips sensors integration and brain computer interface in order to maintain human robot interaction to read the minds of the robots and vice versa.
    - In the manner to integrate this robotics and artificial intelligence method together there must be a hardware platform that just to make the robot to the real time manner, in this regard the powerful computers with GPU or TPU must be available and the python in other side can provide an interactive programming for development of real time AI system in humanoid robot developments.

References

Abiyev et al, H. (2015). Improved Path-Finding Algorithm for Robot Soccers. *Journal of Automation and Control Engineering*, 5.

Agostini et al, A. (2017). Efficient interactive decision-making framework for robotic applications. *Elsiveir Artificial Intelligence*, 187–212.

Akila et al. (2021). Reinforcement Learning For Walking Robot. *IOP Conf. Series: Materials Science and Engineering.* IOP.

Alcaraz et al, J. (2013). Robust feedback control of zmp-based gait for the humanoid robot nao. *The International Journal Of Robotics Research*, 1074–1088.

Arbulu, M. (2009). “Stable locomotion of humanoid robots based on mass concentrated model".

Asimov, I. (1991). *I, Robot.* Spectra.

Boston Dynamics. (2013). Retrieved from CHEETAH - Fastest Legged Robot: http://www.bostondynamics.com/robot_cheetah.html.

Boston Dynamics. (2014). Retrieved from Legged Squad Support Systems: http://www.bostondynamics.com/robot_ls3.html.

Brehm, M. (2017). Constraints and Requirements on the Use of Autonomous Weapon Systems Under International Humanitarian and Human Rights Law. *Geneva Academy of International Humanitarian Law and Human Rights, Academy briefing no. 9*, (pp. 42–68).

Breiman, L. (2001). Statistical modeling he two cultures (with and a rejoinder by the author). In *Statistical science* (pp. 199-231).

C. Graf and T. Rofer. (2010). A closed-loop 3d-lipm gait for the robocup standard platform league humanoid. *IEEE-RAS International Conference on Humanoid Robotics*, (pp. 15–22).

Choi et al. (2004). On the stability of indirect zmp controller for biped robot systems. *in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 2*, pp. 1966–1971. (IEEE Cat. No.04CH37566).

Dekker, M. (2009). *Zero-moment point method for stable biped walkin.* Eindhoven, University of Technology.

Diego .R, & Sven.B. (2021). DeepWalk: Omnidirectional Bipedal Gait by Deep Reinforcement Learning. *In: Proceedings of the International Conference on Robotics and Automation (ICRA) 2021.* IEEE.

Erbatur, K., & Kurt, O. (2009). Natural zmp trajectories for biped robot reference generation. *IEEE Transactions on Industrial Electronics*, 835–845.

Erik et al. (2010). The Design of LEO: a 2D Bipedal Walking Robot for Online Autonomous Reinforcement Learning. *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3228-3237). Taipei, Taiwan: IEEE.

Fogel, D. (2006). *Evolutionary computation: toward a new philosophy of machine intelligence vol. 1.* John Wiley & Sons.

Goswami, A. (1999). "Postural stability of biped robots and the foot rotation indicator (fri) point,". *International Journal of Robotics Research*.

Goswami, A. (1999). "Postural stability of biped robots and the foot-rotation indicator point,". *Int. J. Rob. Res*, 523–533.

Hirai et al, K. (1998). The Development of Honda Humanoid Robot. *in Proc. IEEE Int. Conf. on Robotics and Automations*, (pp. 1321-1326).

Hong et al. (2020). Real-Time Constrained Nonlinear Model Predictive Control on SO(3) for Dynamic Legged Locomotion*. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Las Vegas, NV, USA (Virtual): IEEE.

Horakova, J. (2006). *The (Short) Robot Chronicle (On the 20th Century Cultural History of Robots).* In: Proc. Int. Workshop Robotics in the Alpe-Adria-Danube Region (RAAD).

Hsu, J. (2014). U.S. Navy Tests Robot Boat Swarm to Overwhelm Enemies. *IEEE Spectrum*.

Huang et al, Q. (1999). high stability, smooth walking pattern for a biped robot. *IEEE International Conference on Robotics and Automation.*

Ingrand , F., & Ghallab, M. (2017). Deliberation for autonomous robots: A survey. *Artificial Intelligence*, 10–44.

InteRegele et al, R. (2004). "ProRobot – Predicting the Future of Humanoid Robots". *In: RoboCup 2003:Robot Soccer World Cup VII.* (pp. 366–73). Springer.

István , S. (2016). CERTAIN ASPECTS OF MILITARY APPLICATIONS OF ROBOTICS IN THE FUTURE MISSIONS. *the hadmernok.*

Jun. M et al. (2004). A Simple Reinforcement Learning Algorithm For Biped Walking. *International Conference on Robotics & Automation.* IEEE.

Kagami et al, S. (2002). Online 3D Vision, Motion Planning and Biped Locomotion Control Coupling System of Humanoid Robot : H7. *in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (pp. 2557-2562).

Kajita et al, S. (2003). Biped walking pattern generation by using preview control of zeromoment point. *in 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422),.*

Kalal et al, R. (2019). Finding Shortest Distance Path and Object Detection and Avoidance Using Image Processing and Artificial Intelligence. *IJARIIE*, 281-287.

Katija et al. (2002). A Realtime Pattern Generator for Biped Walking. *international Conference on Robotics and automotion* (pp. 31-38). washigton DC: IEEE.

Katija et al. (2010). Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking. *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 18-22). Taipei, Taiwan: IEEE.

Kim et al. (2019). Dynamic Locomotion For Passive-Ankle Biped Robots And Humanoids Using Whole-Body Locomotion Control. 1-18.

Knight, J. (2002). Safety-critical Systems: Challenges and Directions. *Proceedings of the 24th International Conference on Software Engineering.*

Kunze , L., & Beetz, M. (2017). Envisioning the qualitative effects of robot manipulation actions using simulation-based projections. *Artificial Intelligence*, 352–380.

Lim et al, H. (2002). Online Walking Pattern Generation for Biped Humanoid Robot with Trunk. *in Proc. IEEE Int. Conf. on Robotics & Automation*, (pp. pp 3111-3116).

Liu et al. (2020). Active Balance Control of Humanoid Locomotion Based on Foot Position Compensation. *Journal of Bionic Engineering*, 134–147.

Marc et al. (2018). *Model Predictive Control of Biped Walking on Humanoid Robot.* Cluj-Napoca, Romania : Department of Automation, Technical University of Cluj-Napoca Str. Constantin Daicoviciu 15, 400020 .

Marc, R. (2006). *Legged robots that balance.* Massachusetts Institute of Technology. doi:mitpress.mit.edu/books/legged-robots-balance,

Mathew et al. (2017). *Model Predictive Control of Underactuated Bipedal Robotic Walking.* Department of Mechanical Engineering, Texas A&M University.

Mohammad et al. (2021). Robust biped locomotion using deep reinforcement learning on top of an analytical control approach. *Robotics and Autonomous Systems, Elisiever* , (1-10), (25).

Nishiwaki et al, K. (2002). “Online generation of humanoid walking motion based on a fast generation method of motion pattern that. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2684–2689).

Novacheck, T. (1998). “The biomechanics of running”. In: Gait Posture 7. 77–95.

Perry, J. (1992). *Gait Analysis – Normal and Pathological Function. 3rd ed.* Slack.

Peter, S. (2015). Military Robotics: Latest Trends and Spatial Grasp Solutions. *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*, 9-16.

Peterka, R. (2002). In *Sensorimotor Integration in Human Postural Control 2002* (pp. 1097–1118). In: J. Neurophysiol. 88, 3.

Rajan, K. (2017). Towards a science of integrated AI and Robotics. *Artificial Intelligence*, Elsevier.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: a modern approach (3rd edition):*. Prentice Hall.

S. G, et al. (2015). "A novel biped pattern generator based on extended zmp and extended cart-table model.," . *Inter- national Journal of Advanced Robotic Systems*, 1–15. doi:10.5772/60783

S. Kajita et al. (2001). The 3d linear inverted pendulum model: A simple modeling for a biped walking pattern generation. 239 – 246.

Saeed, A., & Reforgiato, D. (2019). *Path Planning of a Mobile Robot in Grid Space using Boundary Node Method.* Cagliari, Italy: Department of Mathematics and Computer Science, University of Cagliari.

Sakagami et al, Y. (2002). The intelligent ASIMO: System overview and integration. *in Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (pp. 2478-2483).

Sardain, P., & Bessonnet, G. (2004, Sept). "Forces acting on a biped robot. center of pressurezero moment point,". *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 34*, 630–637.

Scianca et al. (2020). MPC for Humanoid Gait Generation: Stability and Feasibility. *IEEE transactions on robotics*, 16-26.

Senn, S. (2007). Statistics in medicine. In S. Senn, *Trying to be precise about vagueness* (p. 1417).

Shafii, N. (2015). *"Development of an optimized omnidirectional walk engine for humanoid robots," PhD diss.* Portugal: University of Porto.

Takanishi et al, A. (1990). Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface. *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, 323–330 .

Tezuka, O. (2002). *Astro Boy.* USA: Dark Horse Manga, Milwaukie.

V.M. (1990). "Biped locomotion: Dynamics, stability, control and application,". *Scientific Fundamentals of Robotics*.

Vukobratovic. (1990). “Biped locomotion: Dynamics, stability, control and application,”. *Scientific Fundamentals of Robotics*.

Wieber, P.-B. (2006). Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations. *IEEE-RAS International Conference on Humanoid Robots.* 38331 St Ismier Cedex, France: IEEE.

Wikipedia. (2021). Retrieved from Center of mass: https://en.wikipedia.org/w/index.php?title=Center_of_mass&oldid=826373631

World Robotics. (2008). VDMA Verlag: International Federation of Robotics (IFR).

World Robotics. (2008). International Federation of Robotics (IFR). VDMA Verlag.

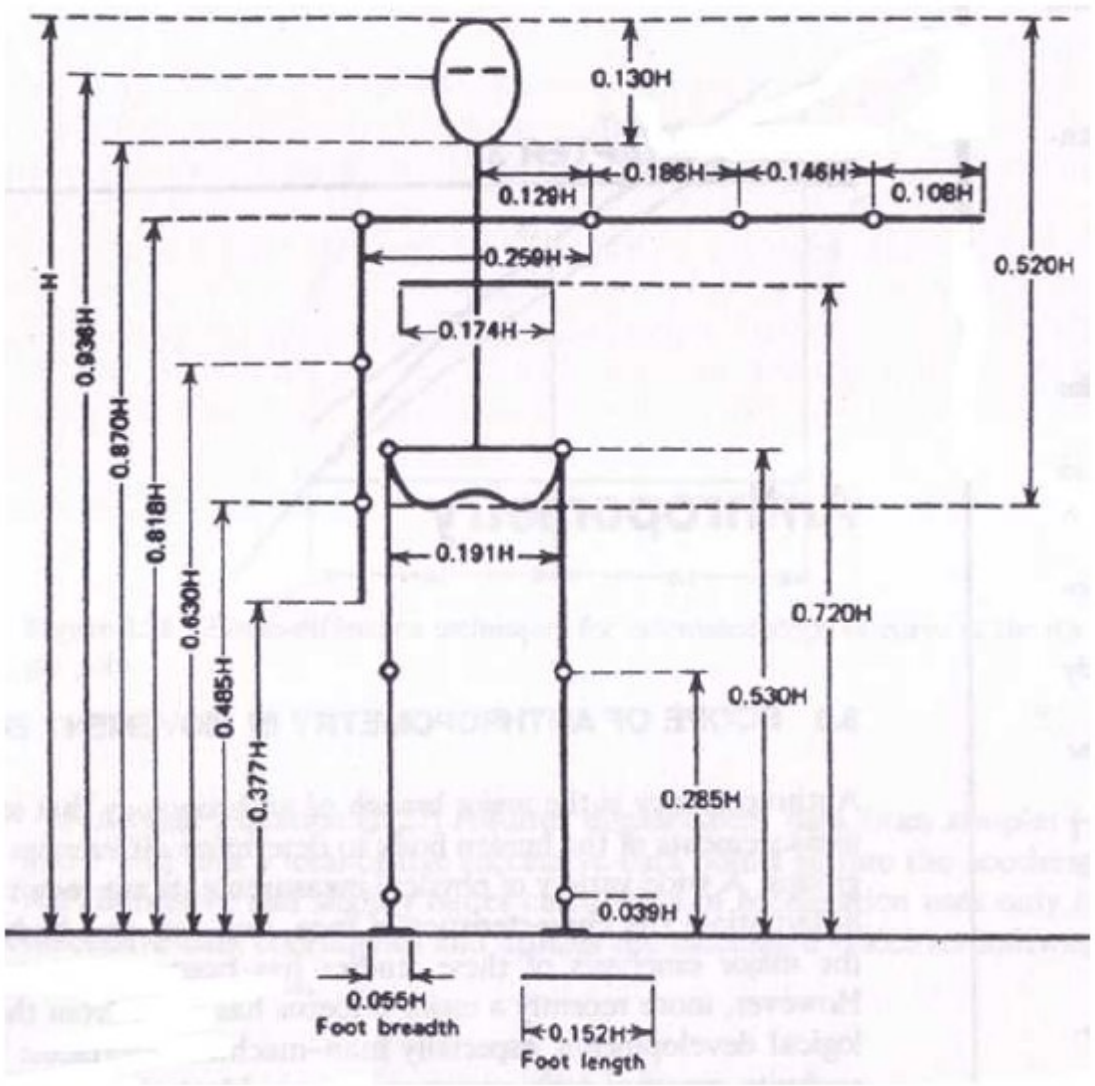
Zadeh, L. (1996). Fuzzy logic computing with words Fuzzy Systems. *IEEE Transactions on, vol. 4*, 103-111.

APPENDIX A

$$x_{knee,R} = x_{ankle,R} + l_1 \cos(\theta_5)$$
$$z_{knee,R} = z_{ankle,R} + l_1 \sin(\theta_5)$$
$$x_{knee,L} = x_{ankle,L} + l_1 \cos(\theta_2)$$
$$z_{knee,L} = z_{ankle,L} + l_1 \cos(\theta_2)$$
$$x_{hip} = x_{ankle,R} + l_1 \cos(\theta_2) + l_2 \cos(\theta_3)$$
$$z_{hip} = z_{ankle,R} + l_1 \sin(\theta_2) + l_2 \sin(\theta_3)$$
$$x_{tor} = x_{hip} + l_3 \cos(\theta_7)$$
$$z_{tor} = x_{hip} + l_3 \sin(\theta_7)$$

$$x_{elbow,R} = x_{tor} + l_{arm} \cos(\theta_{arm,R})$$
$$z_{elbow,R} = z_{tor} + l_{arm} \sin(\theta_{arm,R})$$
$$x_{elbow,L} = x_{tor} + l_{arm} \cos(\theta_{arm,L})$$
$$z_{elbow,L} = z_{tor} + l_{arm} \sin(\theta_{arm,L})$$
$$x_{wrist,R} = x_{elbow,R} + l_{forearm} \cos(\theta_{forearm,R})$$
$$z_{wrist,R} = z_{elbow,R} + l_{forearm} \sin(\theta_{forearm,R})$$
$$x_{wrist,L} = x_{elbow,L} + l_{forearm} \cos(\theta_{forearm,L})$$
$$z_{wrist,L} = z_{elbow,L} + l_{forearm} \sin(\theta_{forearm,L})$$

APPENDIX B: The average Human Body Dimension Height proportion



APPENDIX C Dinkinesh Biped Jacobian in the RCS

I take the robot coordinate system as the base coordinate system. Let d_y = HipOffsetY, d_z = HipOffsetZ and d_{fh} = Foot Height, the Jacobian in the base coordinate system for the left leg is calculated as follows.

$\theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$.

$$z_1^{Base}$$

$${}^{Base}T_1 = {}^{Base}T_0.{}^0T_1 = \begin{pmatrix} s\theta_1 & c\theta_1 & 0 & 0 \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & \frac{\sqrt{2}}{2} & d_y \\ \frac{\sqrt{2}}{2}c\theta_1 & -\frac{\sqrt{2}}{2}s\theta_1 & -\frac{\sqrt{2}}{2} & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$z_2^{Base}$$

$${}^{Base}T_2 = {}^{Base}T_0.{}^0T_1.{}^1T_2 = \begin{pmatrix} \times & \times & c\theta_1 & 0 \\ \times & \times & -\frac{\sqrt{2}}{2}s\theta_1 & d_y \\ \times & \times & -\frac{\sqrt{2}}{2}s\theta_1 & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$z_3^{Base}$$

$${}^{Base}T_3 = {}^{Base}T_0.{}^0T_1.{}^1T_2.{}^2T_3 = \begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_1 + c\theta_2) & 0 \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2\right) & d_y \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2\right) & -d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$z_4^{Base}$$

$${}^{Base}T_4 = {}^{Base}T_0.{}^0T_1.{}^1T_2.{}^2T_3.{}^3T_4 = \begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_2 + c\theta_2) & \times \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2\right) & \times \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2\right) & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$z_5^{Base}$$

$${}^{Base}T_5 = {}^{Base}T_0.{}^0T_1.{}^1T_2.{}^2T_3.{}^3T_4.{}^4T_5 = \begin{pmatrix} \times & \times & \frac{\sqrt{2}}{2}s\theta_1(s\theta_2 + c\theta_2) & \times \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)c\theta_2 + (c\theta_1 - 1)s\theta_2\right) & \times \\ \times & \times & \frac{1}{2}\left((1+c\theta_1)s\theta_2 + (c\theta_1 - 1)c\theta_2\right) & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$z_6^{Base}$$

$${}^{Base}T_6 = {}^{Base}T_0.{}^0T_1.{}^1T_2.{}^2T_3.{}^3T_4.{}^4T_5.{}^5T_6 = \begin{pmatrix} \times & \times & i_{13} & \times \\ \times & \times & i_{23} & \times \\ \times & \times & i_{33} & \times \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Were

$$i_{13} = c\theta_1 c(\theta_3 + \theta_4 + \theta_5) - c\left(\tfrac{\pi}{4} + \theta_2\right)s\theta_1 s(\theta_3 + \theta_4 + \theta_5), i_{23} = \tfrac{1}{2}\left((1+c\theta_1)s\theta_2 + (1-c\theta_1)c\theta_2\right)s(\theta_3 + \theta_4 + \theta_5) - \frac{\sqrt{2}}{2}s\theta_1 c(\theta_3 + \theta_4 + \theta_5)$$

$$i_{33} = -\tfrac{1}{2}\left((1-c\theta_1)s\theta_2 + (1+c\theta_1)c\theta_2\right)s(\theta_3 + \theta_4 + \theta_5) - \frac{\sqrt{2}}{2}s\theta_1 c(\theta_3 + \theta_4 + \theta_5)$$

If the position of the left foot in the RCS is expressed as ${}^{Base}x_p^l = [x \quad y \quad z]$, ${}^{Base}x_p^l$ can be calculated by means of forward kinematics.

$${}^{Base}z_p^l$$

$${}^{Base}T_{End} = {}^{Base}T_0.{}^0T_1.{}^1T_2.{}^2T_3.{}^3T_4.{}^4T_5.{}^5T_6.{}^6T_{End} = \begin{pmatrix} \times & \times & \times & x \\ \times & \times & \times & y \\ \times & \times & \times & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

were

$$x=-\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+\frac{\sqrt{2}}{2}d_{fh}s\theta_1s\theta_6\left(c\theta_2+s\theta_2\right)+$$

$$\left(c\theta_2s\theta_3+\frac{\sqrt{2}}{2}s\theta_1c\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$

$$y=\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+d_{fh}\frac{1}{2}s\theta_6\left(\left(c\theta_1+1\right)c\theta_2\right)$$

$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)+d_y$$

$$z=\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+d_{fh}\frac{1}{2}s\theta_6$$

$$\left(\left(c\theta_1-1\right)c\theta_2+\left(c\theta_1+1\right)s\theta_2\right)+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)-d_z$$

Therefore

$$\frac{\partial x}{\partial\theta_1}=\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+\frac{\sqrt{2}}{2}d_{fh}c\theta_1s\theta_6\left(c\theta_2+s\theta_2\right)$$

$$+\left(-c\theta_1s\theta_3+\frac{\sqrt{2}}{2}c\theta_1c\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)....\frac{\partial x}{\partial\theta_2}=-\frac{\sqrt{2}}{2}s\theta_1c\theta_3\left(c\theta_2+s\theta_2\right)$$

$$\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+d_{fh}\frac{\sqrt{2}}{2}s\theta_1s\theta_6\left(c\theta_2-s\theta_2\right)-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2+s\theta_2\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$

$$\frac{\partial x}{\partial\theta_3}=-\left(c\theta_1s\theta_3+\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)$$

$$\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right).....\frac{\partial x}{\partial\theta_4}=-\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(-d_{fh}s\theta_4s\theta_5c\theta_6+d_{fh}c\theta_4c\theta_5c\theta_6-a_5c\theta_4\right)$$

$$+\left(c\theta_1s\theta_3+\frac{\sqrt{2}}{2}s\theta_1c\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$

$$\frac{\partial x}{\partial\theta_5}=-\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(c\theta_2-s\theta_2\right)\right)\left(d_{fh}c\theta_4c\theta_5c\theta_6-d_{fh}s\theta_4s\theta_5c\theta_6\right)+\left(c\theta_1s\theta_3+\frac{\sqrt{2}}{2}s\theta_1c\theta_3\left(c\theta_2-s\theta_2\right)\right)$$

$$\left(d_{fh}s\theta_4c\theta_5c\theta_6+d_{fh}c\theta_4s\theta_5c\theta_6\right),,....\frac{\partial x}{\partial\theta_6}=\left(c\theta_1c\theta_3-\frac{\sqrt{2}}{2}s\theta_1s\theta_3\left(s\theta_2+s\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5s\theta_6+d_{fh}s\theta_4c\theta_5s\theta_6\right)$$

$$-d_{fh}\frac{\sqrt{2}}{2}s\theta_1c\theta_6\left(c\theta_2-s\theta_2\right)+\left(c\theta_1s\theta_3+\frac{\sqrt{2}}{2}s\theta_1c\theta_3\left(s\theta_2+c\theta_2\right)\right)\left(-d_{fh}s\theta_4c\theta_5c\theta_6+d_{fh}s\theta_4s\theta_5s\theta_6+d_{fh}c\theta_4c\theta_5s\theta_6\right)$$

$$\frac{\partial y}{\partial\theta_1}=\left(\frac{\sqrt{2}}{2}c\theta_1c\theta_3+\frac{1}{2}s\theta_1s\theta_3\left(s\theta_2-c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)-\frac{1}{2}d_{fh}s\theta_1s\theta_6\left(s\theta_2+c\theta_2\right)$$

$$-\left(\frac{\sqrt{2}}{2}c\theta_1s\theta_3+\frac{1}{2}s\theta_1c\theta_3\left(s\theta_2-c\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$

$$\frac{\partial y}{\partial\theta_2}=\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)c\theta_2-\left(c\theta_1-1\right)s\theta_2\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)+\frac{1}{2}d_{fh}s\theta_6\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)$$

$$-\frac{1}{2}c\theta_3\left(\left(c\theta_1+1\right)c\theta_2+\left(c\theta_1-1\right)s\theta_2\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$

$$\frac{\partial y}{\partial\theta_3}=\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1+1\right)s\theta_2-\left(c\theta_1-1\right)c\theta_2\right)\right)$$

$$\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)-\left(\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)$$

$$\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$

$$\frac{\partial y}{\partial\theta_4}=\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(-d_{fh}s\theta_4s\theta_5c\theta_6+d_{fh}c\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$

$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$

$$\frac{\partial y}{\partial \theta_5}=\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4c\theta_5c\theta_6-d_{fh}s\theta_4s\theta_5c\theta_6\right)$$
$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}s\theta_4c\theta_5c\theta_6+d_{fh}c\theta_4s\theta_5c\theta_6\right)$$
$$\frac{\partial y}{\partial \theta_6}=-\left(\frac{\sqrt{2}}{2}c\theta_1c\theta_3+\frac{1}{2}s\theta_1s\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4c\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6\right)$$
$$+\frac{1}{2}d_{fh}c\theta_6\left(\left(c\theta_1+1\right)c\theta_2+\left(c\theta_1-1\right)s\theta_2\right)$$
$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1+1\right)s\theta_2+\left(c\theta_1-1\right)c\theta_2\right)\right)\left(-d_{fh}s\theta_4s\theta_5s\theta_6+d_{fh}c\theta_4c\theta_5s\theta_6\right)$$

$$\frac{\partial z}{\partial \theta_1}=\left(\frac{\sqrt{2}}{2}c\theta_1c\theta_3+\frac{1}{2}s\theta_1s\theta_3\left(s\theta_2-s\theta_3\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$
$$-\frac{1}{2}d_{fh}s\theta_1s\theta_6\left(s\theta_2+s\theta_2\right)$$
$$+\left(-\frac{\sqrt{2}}{2}c\theta_1s\theta_3+\frac{1}{2}s\theta_1c\theta_3\left(s\theta_2-c\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$
$$\frac{\partial z}{\partial \theta_2}=-\frac{1}{2}s\theta_3\left(\left(c\theta_1-1\right)c\theta_2+\left(c\theta_1+1\right)s\theta_2\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$
$$+\frac{1}{2}d_{fh}s\theta_6\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)$$
$$-\frac{1}{2}c\theta_3\left(\left(c\theta_1-1\right)c\theta_2+\left(c\theta_1+1\right)s\theta_2\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$
$$\frac{\partial z}{\partial \theta_3}=\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$
$$-\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}s\theta_4s\theta_5c\theta_6-d_{fh}c\theta_4c\theta_5c\theta_6+a_5c\theta_4+a_4\right)$$
$$\frac{\partial z}{\partial \theta_4}=\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(-d_{fh}s\theta_4s\theta_5c\theta_6+d_{fh}c\theta_4c\theta_5c\theta_6-a_5c\theta_4\right)$$
$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5c\theta_6+d_{fh}s\theta_4c\theta_5c\theta_6-a_5s\theta_4\right)$$
$$\frac{\partial z}{\partial \theta_5}=-\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4c\theta_5c\theta_6-d_{fh}s\theta_4s\theta_5c\theta_6\right)$$
$$+\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}s\theta_4c\theta_5c\theta_6+d_{fh}c\theta_4s\theta_5c\theta_6\right)$$
$$\frac{\partial z}{\partial \theta_6}=-\left(\frac{\sqrt{2}}{2}s\theta_1c\theta_3+\frac{1}{2}s\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(d_{fh}c\theta_4s\theta_5s\theta_6+d_{fh}s\theta_4c\theta_5s\theta_6\right)$$
$$+\frac{1}{2}d_{fh}c\theta_6\left(\left(c\theta_1-1\right)c\theta_2+\left(c\theta_1+1\right)s\theta_2\right)+$$
$$\left(-\frac{\sqrt{2}}{2}s\theta_1s\theta_3+\frac{1}{2}c\theta_3\left(-\left(c\theta_1-1\right)s\theta_2+\left(c\theta_1+1\right)c\theta_2\right)\right)\left(-d_{fh}s\theta_4s\theta_5s\theta_6+d_{fh}c\theta_4c\theta_5s\theta_6\right)$$

Combining the angular and linear Jacobians, we get the Jacobian for the left leg

$$J_{left}^{Base}=\begin{pmatrix}\frac{\partial x}{\partial\theta_1} & \frac{\partial x}{\partial\theta_2} & \frac{\partial x}{\partial\theta_3} & \frac{\partial x}{\partial\theta_4} & \frac{\partial x}{\partial\theta_5} & \frac{\partial x}{\partial\theta_6}\\ \frac{\partial y}{\partial\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\partial y}{\partial\theta_3} & \frac{\partial y}{\partial\theta_4} & \frac{\partial y}{\partial\theta_5} & \frac{\partial y}{\partial\theta_6}\\ \frac{\partial z}{\partial\theta_1} & \frac{\partial z}{\partial\theta_2} & \frac{\partial z}{\partial\theta_3} & \frac{\partial z}{\partial\theta_4} & \frac{\partial z}{\partial\theta_5} & \frac{\partial z}{\partial\theta_6}\\ z_2^{Base} & z_2^{Base} & z_3^{Base} & z_4^{Base} & z_5^{Base} & z_6^{Base}\end{pmatrix} \quad \text{(D.11)}$$

The position of the end-effector of the right leg in the base coordinate frame is expressed as ${}^{Base}x_p^T=[x,y,z]^T$, were

$$\begin{aligned}x=&-\left(c\theta_7c\theta_9-\frac{\sqrt{2}}{2}s\theta_7s\theta_9\left(c\theta_8+s\theta_8\right)\right)\left(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12}+d_{fh}s\theta_{10}c\theta_{11}c\theta_{12}-a_{11}s\theta_{10}\right)\\&+\frac{\sqrt{2}}{2}d_{fh}s\theta_7s\theta_{12}\left(s\theta_8-c\theta_8\right)\\&+\left(c\theta_7s\theta_9+\frac{\sqrt{2}}{2}s\theta_7c\theta_9\left(s\theta_8+c\theta_8\right)\right)\left(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}-d_{fh}c\theta_{10}c\theta_{11}c\theta_{12}+a_{11}c\theta_{10}+a_{10}\right)\\y=&-\left(\frac{\sqrt{2}}{2}s\theta_7c\theta_9-\frac{1}{2}s\theta_9\left(\left(1-c\theta_7\right)c\theta_8-\left(c\theta_7+1\right)s\theta_8\right)\right)\left(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12}+d_{fh}s\theta_{10}c\theta_{11}c\theta_{12}-a_{11}s\theta_{10}\right)\\&+\frac{1}{2}d_{fh}s\theta_{12}\left(\left(c\theta_7+1\right)c\theta_8+\left(1-c\theta_7\right)s\theta_8\right)\\&+\left(\frac{\sqrt{2}}{2}s\theta_7s\theta_9+\frac{1}{2}c\theta_9\left(\left(1-c\theta_7\right)c\theta_8-\left(c\theta_7+1\right)s\theta_8\right)\right)\left(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}-d_{fh}c\theta_{10}c\theta_{11}c\theta_{12}+a_{11}c\theta_{10}+a_{10}\right)-d_y\\z=&\left(\frac{\sqrt{2}}{2}s\theta_7c\theta_9+\frac{1}{2}s\theta_9\left(\left(c\theta_7+1\right)c\theta_8+\left(c\theta_7-1\right)s\theta_8\right)\right)\left(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12}+d_{fh}s\theta_{10}c\theta_{11}c\theta_{12}-a_{11}s\theta_{10}\right)\\&+\frac{1}{2}d_{fh}s\theta_{12}\left(\left(1-c\theta_7\right)c\theta_8+\left(c\theta_7+1\right)s\theta_8\right)\\&+\left(-\frac{\sqrt{2}}{2}s\theta_7s\theta_9+\frac{1}{2}c\theta_9\left(\left(c\theta_7-1\right)s\theta_8+\left(c\theta_7+1\right)c\theta_8\right)\right)\left(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}-d_{fh}c\theta_{10}c\theta_{11}c\theta_{12}+a_{11}c\theta_{10}+a_{10}\right)-d_z\end{aligned}$$

The Jacobian for the right leg is as follows:

$$J_{right}^{Base}=\begin{pmatrix}\frac{\partial x}{\partial\theta_7} & \frac{\partial x}{\partial\theta_8} & \frac{\partial x}{\partial\theta_9} & \frac{\partial x}{\partial\theta_{10}} & \frac{\partial x}{\partial\theta_{11}} & \frac{\partial x}{\partial\theta_{12}}\\ \frac{\partial y}{\partial\theta_7} & \frac{\partial y}{\partial\theta_8} & \frac{\partial y}{\partial\theta_9} & \frac{\partial y}{\partial\theta_{10}} & \frac{\partial y}{\partial\theta_{11}} & \frac{\partial y}{\partial\theta_{12}}\\ \frac{\partial z}{\partial\theta_7} & \frac{\partial z}{\partial\theta_8} & \frac{\partial z}{\partial\theta_9} & \frac{\partial z}{\partial\theta_{10}} & \frac{\partial z}{\partial\theta_{11}} & \frac{\partial z}{\partial\theta_{12}}\\ z_7^{Base} & z_8^{Base} & z_9^{Base} & z_{10}^{Base} & z_{11}^{Base} & z_{12}^{Base}\end{pmatrix}$$

where

Therefore

$$\begin{aligned}\frac{\partial x}{\partial\theta_7}=&\left(s\theta_7c\theta_9+\frac{\sqrt{2}}{2}c\theta_7s\theta_9\left(c\theta_8+s\theta_8\right)\right)\left(d_{fh}c\theta_{10}s\theta_{11}c\theta_{12}+d_{fh}s\theta_{10}c\theta_{11}c\theta_{12}-a_{11}s\theta_{10}\right)\\&+\frac{\sqrt{2}}{2}d_{fh}c\theta_7s\theta_{12}\left(s\theta_8-c\theta_8\right)\\&+\left(-s\theta_7s\theta_9+\frac{\sqrt{2}}{2}c\theta_7c\theta_9\left(c\theta_8+s\theta_8\right)\right)\left(d_{fh}s\theta_{10}s\theta_{11}c\theta_{12}-d_{fh}c\theta_{10}c\theta_{11}c\theta_{12}+a_{11}c\theta_{10}+a_{10}\right)\end{aligned}$$

$$\frac{\partial x}{\partial \theta_8} = \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 - s\theta_8\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$+\frac{\sqrt{2}}{2} d_{fh} s\theta_7 s\theta_{12}\left(c\theta_8 + s\theta_8\right) + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 \left(c\theta_8 - s\theta_8\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial x}{\partial \theta_9} = -\left(c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 - s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$+\left(c\theta_7 s\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial x}{\partial \theta_{10}} = -\left(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - a_{11} c\theta_{10}\right)$$

$$+\left(c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$\frac{\partial x}{\partial \theta_{11}} = -\left(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}\right)$$

$$+\left(c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} s\theta_{11} c\theta_{12}\right)$$

$$\frac{\partial x}{\partial \theta_{12}} = \left(c\theta_7 c\theta_9 - \frac{\sqrt{2}}{2} s\theta_7 s\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} s\theta_{12}\right)$$

$$+\frac{\sqrt{2}}{2} d_{fh} s\theta_7 c\theta_{12}\left(s\theta_8 - s\theta_8\right)$$

$$+\left(c\theta_7 s\theta_9 + \frac{\sqrt{2}}{2} s\theta_7 c\theta_9 \left(s\theta_8 + c\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} s\theta_{12}\right)$$

$$\frac{\partial y}{\partial \theta_7} = \left(-\frac{\sqrt{2}}{2} c\theta_7 c\theta_9 + \frac{1}{2} s\theta_7 s\theta_9 \left(s\theta_8 + c\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$-\frac{1}{2} d_{fh} s\theta_7 s\theta_{12}\left(s\theta_8 - c\theta_8\right)$$

$$+\left(\frac{\sqrt{2}}{2} c\theta_7 s\theta_9 + \frac{1}{2} s\theta_7 c\theta_9 \left(s\theta_8 + c\theta_8\right)\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial y}{\partial \theta_8} = \frac{1}{2} s\theta_9 \left(-\left(c\theta_7 - 1\right) s\theta_8 - \left(c\theta_7 + 1\right) c\theta_8\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$+\frac{1}{2} d_{fh} s\theta_{12}\left(-\left(c\theta_7 + 1\right) s\theta_8 + \left(1 - c\theta_7\right) c\theta_8\right) + \frac{1}{2} c\theta_9 \left(\left(c\theta_7 - 1\right) s\theta_8 - \left(c\theta_7 + 1\right) c\theta_8\right)$$

$$\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial y}{\partial \theta_9} = \left(\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left(\left(1 - c\theta_7\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$+\left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 \left(\left(1 - c\theta_7\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial y}{\partial \theta_{10}} = -\left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 \left(\left(1 - c\theta_7\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - a_{11} c\theta_{10}\right)$$

$$+\left(\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left(\left(1 - c\theta_7 + 1\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$\frac{\partial y}{\partial \theta_{11}} = -\left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 \left(\left(1 - c\theta_7\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}\right)$$

$$+\left(\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left(\left(1 - c\theta_7\right) c\theta_8 - \left(c\theta_7 + 1\right) s\theta_8\right)\right)\left(d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} s\theta_{11} c\theta_{12}\right)$$

$$\frac{\partial y}{\partial \theta_{12}} = \left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 - \frac{1}{2} s\theta_9 \left((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} s\theta_{12}\right)$$
$$+\frac{1}{2} d_{fh} c\theta_{12}\left((c\theta_7+1)c\theta_8 + (1-c\theta_7)s\theta_8\right)$$
$$+\left(\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left((1-c\theta_7)c\theta_8 - (c\theta_7+1)s\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} s\theta_{12}\right)$$

$$\frac{\partial z}{\partial \theta_7} = \left(\frac{\sqrt{2}}{2} c\theta_7 c\theta_9 - \frac{1}{2} s\theta_7 s\theta_9 \left(s\theta_8 + c\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$
$$+\frac{1}{2} d_{fh} s\theta_7 s\theta_{12}\left(c\theta_8 - s\theta_8\right)$$
$$-\left(\frac{\sqrt{2}}{2} c\theta_7 s\theta_9 + \frac{1}{2} s\theta_7 c\theta_9 \left(c\theta_8 + s\theta_8\right)\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial z}{\partial \theta_8} = \frac{1}{2} s\theta_9 \left((c\theta_7-1)c\theta_8 - (c\theta_7+1)s\theta_8\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$
$$-\frac{1}{2} d_{fh} s\theta_{12}\left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)$$
$$+\frac{1}{2} s\theta_9 \left((c\theta_7-1)c\theta_8 - (c\theta_7+1)s\theta_8\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial z}{\partial \theta_9} = \left(-\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$
$$-\left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} s\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} - d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} + a_{11} c\theta_{10} + a_{10}\right)$$

$$\frac{\partial z}{\partial \theta_{10}} = \left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} s\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - a_{11} c\theta_{10}\right)$$
$$+\left(-\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} - a_{11} s\theta_{10}\right)$$

$$\frac{\partial z}{\partial \theta_{11}} = \left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} s\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} c\theta_{10} c\theta_{11} c\theta_{12} - d_{fh} s\theta_{10} s\theta_{11} c\theta_{12}\right)$$
$$+\left(-\frac{\sqrt{2}}{2} s\theta_7 s\theta_9 + \frac{1}{2} c\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} s\theta_{10} c\theta_{11} c\theta_{12} + d_{fh} c\theta_{10} s\theta_{11} c\theta_{12}\right)$$

$$\frac{\partial z}{\partial \theta_{12}} = -\left(\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} s\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(d_{fh} c\theta_{10} s\theta_{11} c\theta_{12} + d_{fh} s\theta_{10} c\theta_{11} s\theta_{12}\right)$$
$$+\frac{1}{2} d_{fh} c\theta_{12}\left((1-c\theta_7)c\theta_8 + (c\theta_7+1)s\theta_8\right)$$
$$+\left(-\frac{\sqrt{2}}{2} s\theta_7 c\theta_9 + \frac{1}{2} c\theta_9 \left((c\theta_7-1)s\theta_8 + (c\theta_7+1)c\theta_8\right)\right)\left(-d_{fh} s\theta_{10} s\theta_{11} s\theta_{12} + d_{fh} c\theta_{10} c\theta_{11} s\theta_{12}\right)$$

$$z_7^{Base} = \begin{pmatrix} 0 \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \qquad z_8^{Base} = \begin{pmatrix} c\theta_7 \\ \frac{\sqrt{2}}{2} s\theta_7 \\ \frac{\sqrt{2}}{2} s\theta_7 \end{pmatrix}$$

$$z_9^{Base} = z_{10}^{Base} = z_{11}^{Base} = \begin{pmatrix} \frac{\sqrt{2}}{2} s\theta_7 \left(s\theta_8 - c\theta_8\right) \\ \frac{1}{2}\left((1+c\theta_7)c\theta_8 + (1-c\theta_7)s\theta_8\right) \\ \frac{1}{2}\left((1-c\theta_7)c\theta_8 + (1+c\theta_7)s\theta_8\right) \end{pmatrix}$$

$$z_{12}^{Base} = \begin{pmatrix} c\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) - s\theta_7 c\left(\theta_8-\frac{\pi}{2}\right) s(\theta_9+\theta_{10}+\theta_{11}) \\ \frac{\sqrt{2}}{2} s\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) + \frac{1}{2}\left((1+c\theta_7)s\theta_8 + (c\theta_7-1)c\theta_8\right) s(\theta_9+\theta_{10}+\theta_{11}) \\ -\frac{\sqrt{2}}{2} s\theta_7 c(\theta_9+\theta_{10}+\theta_{11}) + \frac{1}{2}\left((1-c\theta_7)s\theta_8 - (c\theta_7+1)c\theta_8\right) s(\theta_9+\theta_{10}+\theta_{11}) \end{pmatrix}$$

Note that if the configuration of the leg obtained from joint position sensors during walking is $\theta = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$, the joint angles used for D-H matrices $\theta = [\theta_1 \quad -\theta_2 \quad \cdots \quad -\theta_6 \quad \theta_7 \quad \theta_8 \quad \cdots \quad \theta_{12}]$ due to the different definitions of joint rotational direction between the robot NAO and D-H parameters.