Model Predictive based Trajectory Tracking Control of Quadruped Robot

Desalegn Tegegn Belay

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 Master's in Electrical Power and Control Engineering

(Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

Model Predictive based Trajectory Tracking Control of Quadruped Robot

Desalegn Tegegn Belay

Advisor: Dr. Endalew Ayenew

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 Master's in Electrical Power and Control Engineering

(Control Engineering)

Office of Graduate Studies

Adama Science and Technology University

July, 2023

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**Model Predictive based Trajectory Tracking Control of Quadruped Robot"** is my own work and has not been submitted to any university for similar purpose. The references used in this work are duly recognized by proper citations.

____________________	_____	____________
Name of student	Signature	Date

RECOMMENDATION

I/we, the advisors' of this thesis, here by certify that I/we have read the revised version of the thesis entitled "Model Predictive based Trajectory Tracking Control of Quadruped Robot" prepared under my/our guidance by Desalegn Tegegn Belay submitted in partial fulfillment of the requirements for the degree of Master's of Science in electrical power and control engineering (control engineering). Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

____________________	__________	____________
Major Advisor	Signature	Date
____________________	__________	____________
C0-advisor	Signature	Date

APPROVAL SHEET

I/we, the advisors of the thesis entitled "Model Predictive based Trajectory Tracking Control of Quadruped Robot" and developed by Desalegn Tegegn Belay, 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 Desalegn Tegegn Belay have read and evaluated the thesis entitled " Model Predictive based Trajectory Tracking Control of Quadruped Robot " 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 (control engineering).

Chairperson	Signature	Date
Internal examiner	Signature	Date
Beteley Teka Hailu (PhD)		July 12, 2023
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 of all I would like to thank GOD, who gave me soul of existence and the strength to survive in this uncomfortable world. Secondly, I would like to thank my family who helped me throughout my life without any boredom. Lastly, I would like to thank my advisor Dr. Endalew A. for his advice and support with his constant guidance.

CONTENTS **PAGE**

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

Abbreviations	description
DOF	Degree of freedom
D-H	Denavit-Hartenberg
MPC	Model predictive control
VMC	Virtual mode control
PD	Proportional derivative
TBC	Trunk balance controller
SLC	Swing leg controller
COM	Centre of mass
PID	Proportional integral derivative
PSO	particle swarm optimization

ABSTRACT

Quadruped robots are four legged robots which has broad applications in complex situations such as military investigation, inspection, planet exploration and so on. The problems with quadruped robots are difficulties to keep their balance during standing and movements. These have been studied and somehow fixed by the researchers and now the difficulty is improving the performance of the robots. So, for the improvement of the trunk centre of mass trajectory tracking accuracy plus to provide proper leg angles behavior for the trajectory of the leg, an efficient control method is required. This thesis presents model predictive based tracking control for the trajectory of quadruped robot. The method proposed involves developing a dynamic model for the quadruped robot to determine the ideal leg contact force that tracks the center-of-mass trajectory. Both linear and circular trajectory was given for the trunk system as a reference input so that it will track it by the provision of proper controller design parameters. The basic parameters of the controllers are tuned with help of a global metaheuristic optimization algorithm called PSO method. As simulation result shows in linear trajectory the steady state error is 9.7702e-05m or 0.0097702%. In general, the proposed controller have met the objective of the study which is important to make the robot perform different given tasks with low time period and with high efficiency.

Keywords:- *CoM, Efficiency, MPC, Optimization, PSO, Quadruped Robot, Trunk Balance*

CHAPTER ONE

1 INTRODUCTION

1.1 Historical Background

Robotics is a modern technology field that is relatively young and crosses engineering traditional boundaries. Understanding robots applications and dealing with their complexity requisites knowledge of mechanical, electrical, industrial and systems, mathematics, economics and computer science engineering sectors. The robotics field and factory automation has become increasingly complex, leading to the development of new engineering disciplines like applications, knowledge, and manufacturing engineering.

A machine that is engineered to carry out tasks automatically with a specified level of speed and accuracy is referred to as a robot. There are many distinct types of robots as based on their requirement to perform different tasks. It is a multifunctional, automatic, re-programmable industrial machine designed to fully or partially replace human in different hazardous environments of works. It can be applicable for the purposes including an automatic car for a child to play with, an automatic machine sweeper, for exploration in space, for different services in military etc.

The desire of developing a legged machine which have the ability to replace human had existed for hundreds of years. Many researches and studies had also been done on it. However, in the last half century, this goal has become achievable, due to the emerging of advanced technology. The two primary classifications for legged robots are static and dynamic machines.

1.1.1 Static Machines

This is a type of legged robots in which the machine COM lies within a polygonal support provided by feet support to fix the stability problem and by sustaining the speed low enough for the minimization of dynamic effect. This approach or the static locomotion requires at least four legs of the robots must touch the ground, so that the three legs provide supporting while one moves. The reduction of number of legs to two can be possible by organizing the feet to provide a base of support. In earliest machines employment of fixed gait was considered for coordination of their legs due to lack of computer technology. That means the legs motion following the same repeating pattern. The early attempts marginally increased the mobility however there is reduction of efficiency and speed. With the advancement of computer based control, the utilization of fixed gait was totally deserted. Robots like Dante II (D. Wettergreen, 1995) had successfully applied a partially fixed gait. Dante II an eight-legged robot, which in July 1994 descended successfully into mount spur volcano in Alaska (Figure 1-1).

Figure 1-1 The Dante II robot

Due to more sophistication of control methods and computers during then 1990', two legged robots or bipeds were developed that can negotiate stairs through static gaits.

1.1.2 Dynamic Machines

Robots that walk using fixed gaits can utilize many important advantages of a legged locomotion. But, they maintain stability by necessarily being slow, many times by employing large number of legs which adds the complexity of mechanical structure. An alternative method is dynamic movement, where the erect body functions akin to an inverted pendulum and requires ongoing regulation to maintain stability. This mode of operation is employed by all mammals while performing a very slow walk. Depending on shaky system dynamics, with fewer legs there is a possibility of increased mobility. However, this requires higher cost and more complex control for active balance. The initial development of an actively balanced hopper was achieved by Matsuoka in 1980, followed by the development of the first 3D actively balanced robot by Miura and Shimoyama in 1981. This bipedal or two-legged robot walked on rigid legs that resembled a human on stilts.

At MIT, Raibert was a key contributor to the field of dynamic legged locomotion, as shown in Figure 1-2. In the 1980s, he designed hopping robots with one, two, and four legs that could maintain balance and move on level ground using a three-part control algorithm with straightforward variations. Ongoing development is underway for quadrupeds that can perform dynamic walking. One such example is Hirose's Titan VI (S. Hirose, 1995), which was constructed in the mid-1990s and featured four actuated degrees of freedom per leg, as well as an additional actuated degree of freedom in the back to aid in climbing stairs. Another example is Scamper (Figure 1-3), also developed in the mid-1990s, which utilized an actuated rotary knee and hip to achieve a bouncing gait.

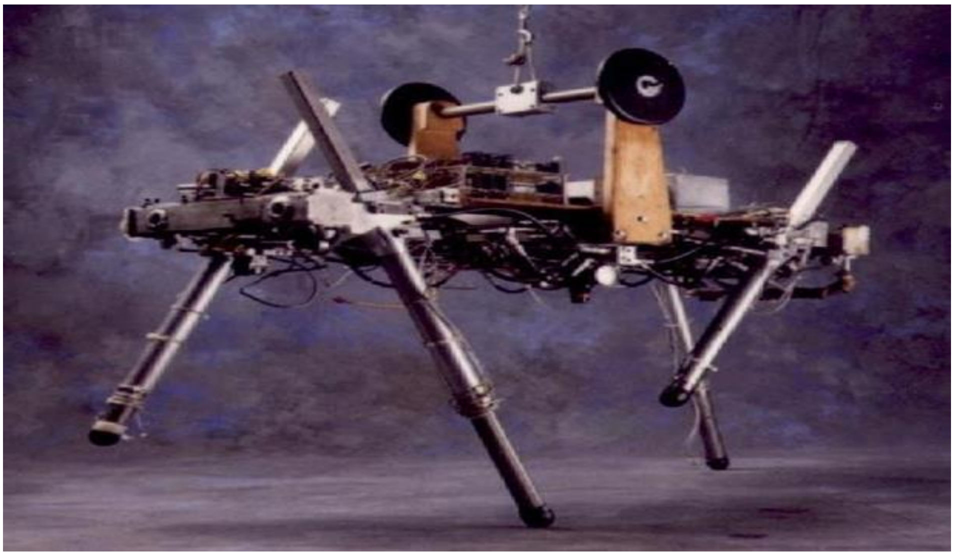

Figure 1-2 The MIT Quadruped

1.2 Legged Robots

The scientific world of inventions is inspired by different wonderful creatures that are present in nature. The curiosity of Human beings to understand and create robots that can imitate the behavior of creatures in nature have opened a door to develop different types of robots that performs different specific tasks. Therefore, there are a number of robots that move themselves by using wheels or by devices that resemble legs. The legged robots are classified by the leg count; they are two legged or bipeds like humans or birds, four legged robots or quadruped like mammals and reptiles, six legged robots or hexapods like insects, and eight legged robots or octopods like spiders. There are so many advantages of a legged robot over a wheel based robot or tack-based robot. Rough and highly uneven terrain cannot be too much obstacles for legged robots while for wheeled robot there must be continuous path to move or to perform a certain given tasks.

Therefore legged robots are suitable to overcome the situations like gaps, soft ground, steps or any other obstacles. One of the type of walking or legged robots that can have the above mentioned feature and considered in thesis work is called quadruped robot, which is four legged robot.

1.3 Quadruped Robots

Many improvements have been made for legged robots in the past few years, however, legged robots with four legs are a particular research topic for a couple of reasons. Various locomotion modes can be utilized by quadruped robots such as trotting, walking, galloping, pacing etc. A huge numbers of these locomotion modes of the robot are still unstudied and potential utilization of them can be used for the improvement of efficiency and increment of the speed locomotion of robotic systems.

Figure 1-3 spot mini and big dog *(Choi, 2021)*

The quadruped robot developed featuring twelve (12) degrees of freedom (DOF) is designed to perform tasks that are highly dynamic like running and jumping. The joints of robots are powered either by electric actuators or by hydraulic actuators, relying on their particular property and performance requirements. As it can be known from the name a quadruped robot means a legged robot with four legs. Each leg should have a minimum of two degrees of freedom. It can have also more than two (2 DOF) degrees of freedom. The quadruped robot main criteria are that the standing still must be passively stable. Quadruped robot can also run and walk. A gait pattern is necessary to move a quadruped robot from one specific area to another. Location of the placing and the coordination of time and raising of each feet while the body moves in its six DOF is called as gait. The main challenge that makes difficulty in controlling the quadruped robot is to maintain stability while shifting body actively during the movement. There are two types of quadruped robot locomotion, the reptilian type and the mammalian type. Locomotion of mammalian type can move faster due to ease in the transition of the leg joints geometrical configuration.

Examples of locomotion which are mammalian based are robotic cheetah, ABIBO robotic dog, etc. Robots which are reptile inspired have a slower speed but higher stability. Heavy loads can be carried by the quadruped robot for military and civilian purposes. These kind of robots are self-balancing, with the use of sensors. The reason why legged robots are chosen over tracked robots is since they are more accessibility than tracked robots. Moreover, legged robots have the capacity to overcome any obstacle or barrier in its path which is not possible for wheeled or tracked robots since they need a continuous path to make a movement.

1.3.1 Classification of Quadruped Robot

There are many ways to classify walking robots.

1. Legs number
2. By a shape of body
3. Locomotion technique
4. Number of degrees of freedom per leg

1.3.2 Advantages of Quadruped Robots

The advantages of quadruped robots are listed as follows:

- Active suspension
- Environmental footprint.
- Mobility – robots with legs are systems that are intrinsically omnidirectional.
- Overcoming obstacles –robots with legs can overcome obstacles like hole, rock, etc.
- Energy efficiency

1.3.3 Disadvantages of Quadruped Robots

- The major issues in legged robots is stability.
- More expensive due to complexity.
- They are more complex than the wheeled one in terms of the machine as well as control and electronics.
- Have more complex control algorithm than wheeled robots.

1.3.4 Applications of Quadruped Robots

- Nuclear power plants inspection
- Civil projects
- Exploration of land, planetary and submarine.

- Military service applications
- Help for disabled people
- Agricultural and forestry tasks.
- Study of living creatures

1.4 Problem Statement

The major problem of legged robots or quadruped robots was maintaining stability or balance control when the robots performs a movement and also when they stands. However, many researches have been done to fix this problem the accuracy and the performance of the robots has become another problem to make a research too. Due to this, some control system theories are utilized to enhance the performance of a quadruped. But, these are not enough to make the robot perform a given task efficiently. So, further improvement of accuracy is required by incorporating suitable technique and optimization algorithm. Therefore, the focus of this thesis is to address the issues of control and accuracy when tracking the movement path of four legged robots, and an optimization based combination of model predictive and PD control methods are proposed as a solution.

1.5 Objective

1.5.1 General Objective

- ✓ Trajectory tracking of quadruped robot with increased accuracy based on model predictive control algorithm.

1.5.2 Specific Objective

The specific objective to be met in this thesis are listed as shown below:

- ✓ Design model predictive controller for the system
- ✓ Optimally tuning the parameters of the proposed controller.
- ✓ Simulate the proposed technique on MATLAB command line.
- ✓ Draw relevant conclusions and recommendations.

1.6 Scope of the Thesis

The work of constructing or developing any type of legged robot with dynamic stability was undoubtedly a difficult one due to different reasons including from difficulties to get the components to be assembled up to cost limitation. This thesis is all about designing of model predictive controller for quadruped robot system. So, the work deals from theoretical studies, mathematical model, code, up to simulation.

1.7 Limitation of the Thesis

Due to the cost and time constraints involved in obtaining the necessary materials deploying is difficult, this thesis will primarily limited to focus on the mathematical design and simulation of the system using MATLAB software.

1.8 Significance of the Thesis

The contributions of this thesis work are organized as follows:

(1) Optimization of the parameters of a MPC algorithm designed to accurately track the trajectory of a quadruped robot while taking into consideration the robot's physical limitations.

(2) Ensuring increased accuracy and performance of quadruped robot through the suitable MPC algorithm.

(3) Contribution to the area of legged locomotion: The thesis contribute to the broader area of legged locomotion by demonstrating the feasibility of using MPC for controlling the motion of quadruped robots. Legged locomotion is a difficult issue due to the complex dynamics and nonlinearity of legged systems, and any progress in this area can have significant implications for robotics applications such as search and rescue, agriculture, and exploration.

(4) Comparison with other control methods: The thesis compare the proposed MPC algorithm with other existing control methods for quadruped robots, such as proportional-integral-derivative (PID) control.

1.9 Organization of the Thesis

The present thesis is structured into six chapters, each addressing a specific aspect of the research. Chapter 1 introduces the background, objectives, problem statement, and significance of the study. Chapter 2 provides a comprehensive literature review of important related works. Chapter 3 outlines the methodology, including the mathematical model and materials necessary for conducting the research. Chapter 4 focuses on the proposed controller design for the system plant. In Chapter 5, the simulation findings and their analysis are presented. Lastly, Chapter 6 concludes the thesis and offers recommendations for future work.

CHAPTER TWO

2 LITERATURE REVIEW

2.1 Quadruped Robot

2.1.1 History

Enterprises with different supplies of energy and actuation systems have made quadruped robots for the past recent decades. The efficiency of both the actuation and the energy systems have also been improved. The first quadruped robots were in a hydraulically actuation form and now through passing different progress most recently robots are in the form of electric one.

2.1.2 Hydraulic Robots

The Big Dog Boston Dynamics (M. Raibert, 2008) shown in Figure 2-1 was one of the first hydraulically actuated type of quadruped robot. In this robot control Inverted Pendulum approach was used. In each leg the robot had four (4) Degrees of Freedom and a gasoline system is used as its source of energy. Due to high maintenance requirements and a bulky motor, it was halted in 2015.

Figure 2-1 The Robot called Big Dog *(Aranda, 2022)*

Advancements in technology and growing interest in robotics led to the development of battery-powered supplies and brushless motors for robots, resulting in improved movement capabilities and increased energy efficiency. The enhanced energy efficiency paved the way for the development of numerous electrically actuated robots by several companies, which employed complex control techniques in their designs.

2.1.3 Electric Robots

With the name of MIT Cheetah, One of electrically actuated robots which uses the hierarchical locomotion was developed in the MIT (B. Katz, 2019) shown in Figure 2-2. In each leg it has three motors, providing 3 DOF per leg and movements of a wide range. In response to its small-inertia, the robot can execute fast and dynamic maneuvers, such as complete backflips.

Figure 2-2 MIT cheetah *(Aranda, 2022)*

Most recently, electrically actuated robot was innovated Boston Dynamics adding another feature, incorporating in its robot spot, an arm on the robot top having five(5) DOF which is capable of manipulating and grabbing objects (Aranda, 2022). This arm can perform autonomous navigation, execute complex movements, and manipulate a doorknob to open a door, among other tasks.

Although there are various control techniques available for robots, model predictive control is the dominant technique that has been widely employed in industrial robotics. This control method has been selected because it’s a successful and suitable method in both the simulations and real robots control. In the thesis work this efficient method have also been selected to meet the objective stated of the work and for eliminating the problem faced in quadruped robot environment.

2.2 Model Predictive Control

In the late 1970s, the foundation of MPC theory was laid independently. In 1978, Model Predictive Heuristic Control (MPHC) was developed, which incorporated the features of MPC:

- a receding horizon,

- The controls iterative determination.
- An explicit process model
- Input and output constraints, and

However, optimal control did not claimed. Instead, an iterative determination of future controls where performed until meeting the constraints. This technique was applied to MIMO systems in the process industry, which are characterized by long processing times and significant delays (Max Schwenzer, 2021).

Model Predictive Control was initially developed to regulate dynamic systems that have multiple inputs and outputs, and are subject to constraints, particularly in chemical applications (Ruchika, 2013). At each sampling time, this control method involves computing the current control action through the solution of a finite horizon open-loop optimal control problem. The problem utilizes the current state of the plant as the initial state and generates an optimal control sequence, with the first control in the sequence being applied to the plant.

MPC depends on models. MPC enables the use of long-established knowledge, eliminating the need for experts to formulate explicit control laws, which can be a tedious task. The control law is automatically determined through optimization, based on a model. This approach offers several advantages, including flexibility, an implicit formulation, and the explicit use of models, which make a strong case for promoting MPC within the engineering community (Max Schwenzer, 2021).

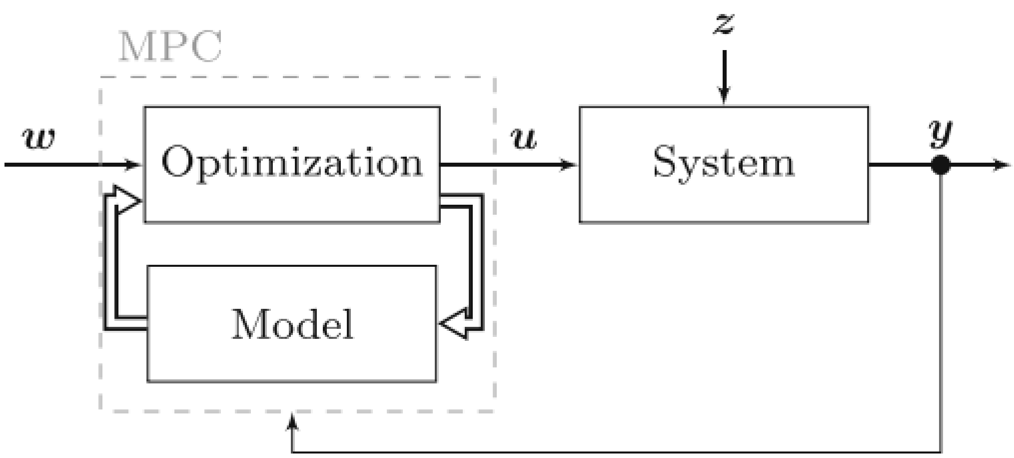

Figure 2-3 MPC-based control loop Simplified block diagram

2.2.1 Theory

MPC is a sophisticated control technique that incorporates a model of the system's future behavior. This model is utilized to solve a constrained optimization problem, which considers the anticipated behavior and results in an optimal output u. This control technique directly integrates constraints, and the cost function is designed to ensure that the system output y follows a reference r over a horizon of N_2, as illustrated in Figure 2-4. Only the first value of the optimized output trajectory is utilized for the system, and the optimization and prediction procedure is repeated at each time step. Hence, MPC is commonly referred to as "receding horizon" control. Unlike other conventional control approaches, MPC combines prediction and optimization, recognizing that long-term optimization may not achieve optimality over a short time due to differences in forecast errors at different time horizons.

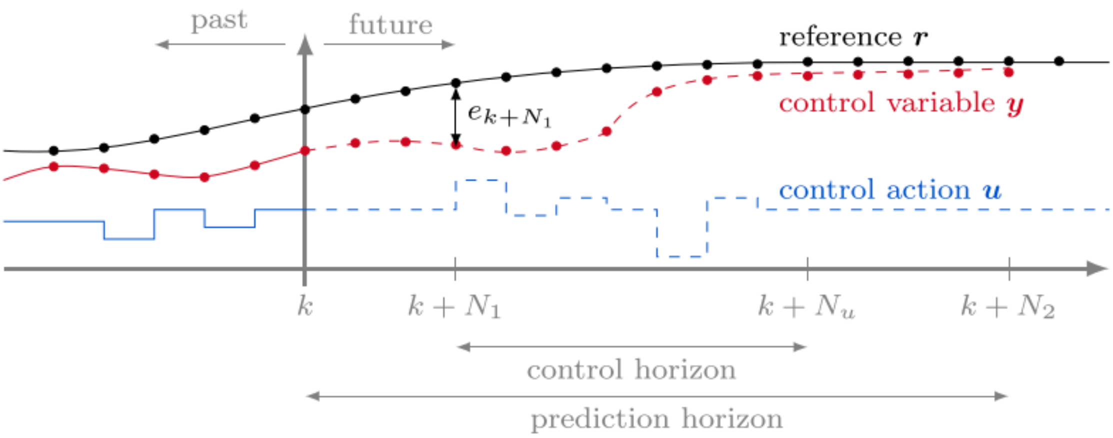

Figure 2-4 Functional principle of predictive model-based control using horizons

The influence of a manipulated variable u on the control variable y is depicted by the prediction horizon N_2, which should be sufficiently long. Delays can be accounted for by the lower prediction horizon N_1. Typically, N_1 is set to 1 to accommodate for computation time. MPC allows for the achievement of high-level objectives, rather than just machine tool set points.

2.3 Review of related works

There are not too many related works that are done with the aim of designing a controller for quadruped robot trajectory tracking control system. Some of the most important papers reviewed to work this thesis are listed as follows and analyzed one by one.

Having a good terrain adaptability makes the quadruped robot suitable for many applications. Complex and uncomfortable situations for human being are difficult for quadruped robot with good feature tasks like planets exploration, military investigation and disaster relief and rescue and so on (Choi, 2021) (Rodino, et al., 2020). Tasks that cannot be performed by the wheeled and tracked robots can be performed by quadruped robot. The control mechanism for quadruped robots comprises of several components, including task oriented decision-making, environmental sensing, path planning, and control of trajectory tracking (Kang, et al., 2021). Of all these components, a robust and effective control of trajectory tracking mechanism is fundamental for four legged robots to achieve comprehension, practicality, and overall functionality. Compared to the focus regarding trajectory monitoring methods in the area of unmanned vehicles, there has been relatively limited research and literature on motion path tracking technology for four legged robots.

The paper owned by (Kalyoncu, 2019) outlines the design and optimization of a hybrid LQR-PID control approach for regulating the foot trajectory of a quadruped robot. In this study, the proposed controller parameters tuning for the foot trajectory control is performed by GWO technique and also compared with two metaheuristic optimization techniques PSO and GA. The two weight matrices of LQR controller and the three PID controller gains are tuned by the proposed optimization algorithms. The design process involves importing a solid model into SIMULINK/simMechanics and obtaining a mathematical model in the form of a state space model. The model is linearized using linear analysis tools, taking into account the leg step motion in the sagittal plane. Generally, from the simulation of the system, GWO technique have showed its superiority than other compared optimization techniques. This paper focuses on the linearization of the system model and the presentation of the leg trajectory, without incorporating the complete dynamic model of the four legged robot. Additionally, the robot leg is constrained to only two degrees of freedom.

An essential metric for evaluating the performance of a robot is achieving a well-balanced state. Planning the ground reaction force (GRF) of quadruped robot's legs during the support phase of locomotion is crucial for maintaining stability and balance. (ChengYe Wu, 2020)

Employing a model predictive control algorithm, the balance control of a quadruped robot was achieved through the planning of ground reaction force (GRF) for position and attitude control. Once accomplished, the forces are then translated into the operational space by means of the joint space Jacobian matrix to acquire the joint space torque. This study utilized Recurdyn and MATLAB for co-simulation purposes and yielded favorable results. The viability and effectiveness of the algorithm were assessed through a comparison with the traditional balance controller. This study establishes the foundation for the design and control of future motion planning for quadruped robots. In this paper work, the body position centroid in the z axis is the only control variable considered.

(Zhang, Rong, Hui, Li, & Li, 2016) And (Zhang, Rong, Li, Cai, & Li, 2016) Proposed a method for trajectory tracking control for trot diagonal walking style of a four legged robot, built upon a virtual model. The method involves setting up virtual model controls for both the support (stance) phase and the swing phase. However, the model only considers static analysis of the system, neglecting the dynamics of the robot. While this make easier the model complication, it leads to in significant tracking control errors. Additionally, the virtual control method relies on a PD controller, which is a direct and intuitive control method that is fast and does not request complicated calculations. However, this approach cannot guarantee accurate control. (Luo, 2022).

The paper (Ming Lu, 2022), presented a design for a modular and lightweight quadruped robot with 2 DOF in the legs. The legs are designed using a horizontal layout to ensure accurate transmission and minimize weight. Each actuator rotation angle is determined using an inverse kinematics algorithm, and the foot-end trajectory is planned to minimize the contact between the foot-end effector and the ground during both the swing and support phases. The robot's gait is designed to accommodate standing, trotting, take-off, and walking. The efficacy of the foot-end trajectory and the stability of the gait are verified through experiments within a prototype. However, this leg configuration only has two degrees of freedom and does not include a center of mass dynamics analysis.

Considering the delay phenomenon in control signals that can occur with PID algorithms, MPC algorithms have the ability to account for this delay in the model and can effectively avoid this problem. (Dini & Majd, 2019). (Carlo, Wensing, Katz, & Kim, 1-5 october,2018) And (Bledt, et al., 1-5 october,2018) proposed a motion control method for a quadruped robot built upon theory of MPC, aimed at determining the ground force of reaction during movement. However, the method only focuses on controlling the trunk and neglects the effect of leg control coupling on the total control of the quadruped robot system. As a result, there is a significant steady-state error in the leg trajectory, which negatively impacts the accuracy of tracking trajectory and stability of the quadruped robot.

In the paper (Ammar A. Aldair, 2019), the authors presented a design for a four-legged robot with intelligent controllers and conducted simulations to evaluate its performance. The robot's architecture includes twelve servo motors, with three in each leg, providing flexibility in movement and turning. To stabilize and control the robot's motion, Proportional Integral Derivative (PID) and Fuzzy controllers were proposed. An ant colony optimization algorithm was used to tune the controllers' parameters. The entire architecture was implemented using the Simscape Multibody package in MATLAB, which is commonly used to model physical systems. Simulation results demonstrated good performance in terms of elapsed time minimization and motion improvement. The Fuzzy controller's stability was examined using the Lyapunov criteria to validate its overall performance. However, this work only focuses on the control of the quadruped robot's legs and does not include the dynamics of the entire robot.

Intelligent control technology over the last few decades, particularly artificial neural networks (ANNs), has undergone significant advancements. (Luan, Na, Huang, & Gao, 2019). Several studies have demonstrated the importance of neural networks in addressing uncertainty problems and function approximation. (Zhang, et al., 2022).

(Hwangbo, et al., 2019) Utilized deep reinforcement learning to increase the performance of a medium-sized quadrupedal robot, specifically the ANYmal robot. Trained policies were applied, resulting in significant improvements in the robot's speed during simulation. (Ma, Liang, & Tian, 2020) Proposed a gait planning method for quadruped robots built upon central pattern generators (CPGs) using the Hopf oscillator to construct control networks for two CPGs. The structures of the CPG networks were modeled and analyzed, and they were able to control each joint of the quadruped robot to attain rapid gait planning. However, the algorithm is slow to converge and overly complex, making it difficult to apply in practical engineering settings.

In the paper (Yulong You Z. Y., 2022) a new practical trajectory tracking control method has been developed to enhance the tracking precision of centre of mass path and foot end point path in a four legged robot entirely powered by electricity. The method comprises of two controllers: the TBC and the SLC. The TBC employs a dynamics model of the quadruped robot to calculate the optimal force at the foot-end, which tracks the CoM trajectory through the use of model predictive control (MPC) principles. On the other hand, the SLC plans a Bessel curve and tracks it using a virtual spring-damping element while implementing the radial basis function neural network (RBFNN) in supervisory control. The presented work has the limitation in eliminating the fluctuation in errors which impacts accuracy and performance of the system.

(PP Jain, 2021) Made a twelve Degree of freedom Quadruped Robot model using Fusion 360. Then, after exporting the model into ROS by using available plugin Fusion2URDF by converting into executable ROS, the simulation on ROS Melodic running is done on Ubuntu18.04LTS. In the simulation the Trot gait is used. After the simulation is done connected with Arduino and then interfacing it with ROS, the same is being implemented on Servo Motor. Graphs like the joint angle, velocity is plotted using Plot Juggler. The paper didn't take into account of the whole quadruped robot trajectory instead it is limited to the leg part of the robot.

In legged robot a dynamic locomotion has a crucial role for fulfilling tasks in different unstructured environments. (Wei Yan, 2021) Proposed a dynamic modeling and whole-body kinematic method for a quadruped robot based on screw theory, which can accommodate different mechanisms and gait topologies. In contrast to simplified models like inverse pendulum or centroid models, the proposed methods have the capability to manage the 10-dimensional inertia and mass for each component of the robot.

The authors considered spherical joints for the foot contact model and formulated three different mechanism topologies for standing, walking, and floating phases. Control strategies were developed for walk and trot gaits, and two additional sources of information were incorporated into the control system: force sensors installed under the feet to provide contact forces and an inertial measurement unit (IMU) to determine body poses. The results demonstrated that the models can achieve steady locomotion and are computationally efficient, but the kinematics and dynamics of the system are complex and bulky, making them difficult to comprehend.

In summary, centered on the above interpretation, the subsequent points represent the most common deficiencies observed in path tracking for quadruped robots:

- Often, trajectory tracking controllers for quadruped robots are based solely on kinematic models, neglecting the system's dynamic characteristics and potentially hindering accuracy improvements.
- Trajectory tracking typically involves a closed-loop simple control strategy, which may not be easily generalized for quadruped robots with multi-variable, under-actuated, nonlinear, and strongly coupled behaviors.
- Existing research on control of quadruped robot trajectory tracking often focuses on trunk centroid trajectory tracking, ignoring the impact of leg trajectory tracking accuracy on the whole control system.
- While the MPC algorithm has been a successful solution for trajectory tracking in unmanned vehicles, further verification is needed to determine if it satisfies the performance criteria for controlling the trajectory tracking of a quadruped robot.

All the journal papers mentioned above are related to the work in this thesis, and all of these important papers referred for the purpose of designing or simulation of the system. Apart from these mentioned papers, some more other journal papers were referred for the work that was helpful. So, to address the control problems described above and improve the accuracy of trajectory tracking for quadruped robots, this thesis proposes a suitable and efficient control method.

CHAPTER THREE

3 METHODOLOGY

3.1 Introduction

In this section the materials required and the methods to be followed are discussed. So, Materials used to work this thesis are mainly Software tools. The software tools required and utilized are:

- MATLAB software and,
- CasADi,

For simulation and analysis of the system model, I utilized MATLAB software, which is a software for general purpose computing with a broad assortment of toolkits. Additionally, CasADi, an open-source software tool utilized for algorithmic differentiation and nonlinear optimization, was employed to enable fast and excellent execution of various optimal numerical control methods, for Non Linear Model Predictive Control (NMPC).

3.2 Modelling of the Quadruped Robot

3.2.1 Kinematics Model

The mechanism of legged walking robot is very complex due to the composition of many DOF. Before simulation, it is necessary to perform kinematics modeling to compute position and orientation of the foot end. The kinematics model specifies the velocities, positions of the legs, and joint variables, which can be used to calculate the torques and forces at the joint, body, and ground of quadruped robots. To obtain the model of kinematics, the quadruped could be represented just like a solid base and 4 legs robot, with every leg having 3 DOF (Bledt., 2018). The robot each leg is comprised of three joints or motors that generate torques to facilitate locomotion for the entire body. With legs accounting for only about 10% of the robot's mass, their inertia is relatively low, enabling the joints or motors to quickly move the legs through the air.

Most quadruped robots are designed as a systems of floating base, consisting of a rigid linked bodies connected to joints of the shoulder and hip. The robot's kinematics can be represented as a tree structure, with links that correspond to each part of the robot. Essentially, a floating base model can be used to describe the quadruped robot with 4 legs, with each leg visualized as a chain of kinematic, as shown in Figure 3-1.

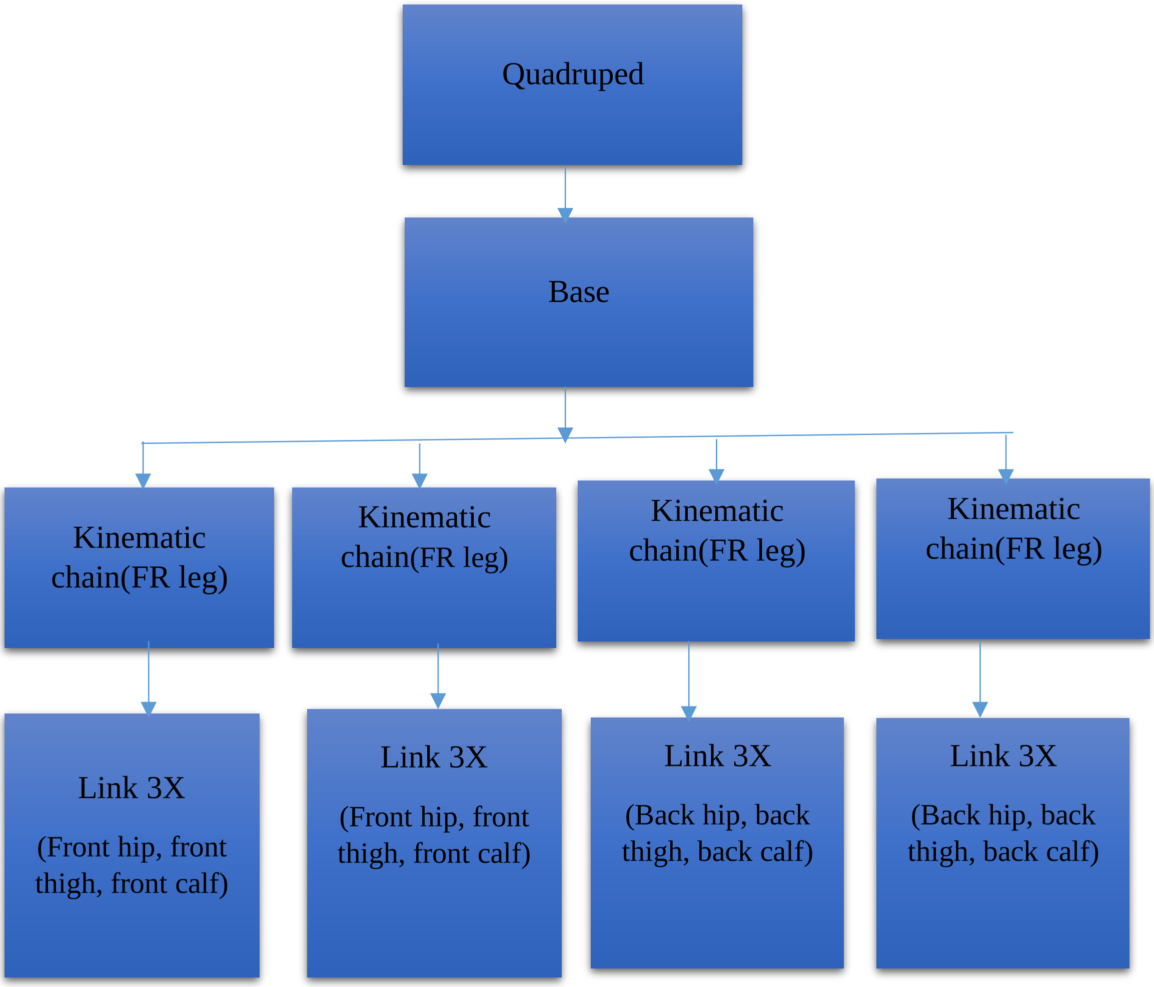

Figure 3-1 The structure of the quadruped robot *(Aranda, 2022)*.

In the context of the robot's perspective, two frames are taken into account:

- The world frame
- The body frame

In Figure 3-2, the body and world frame are depicted, with the former having a reference outside the robot body and the latter having its reference at the robot's COM. When manipulating or visualizing the robot in the real world or simulation, the user typically interacts with it from the perspective of the global frame, whereas the local frame primarily refers to the quadruped robot itself.

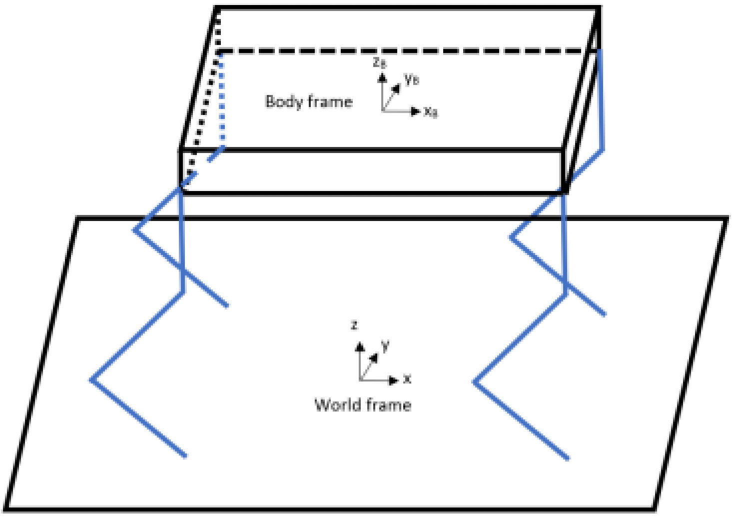

Figure 3-2 Quadruped robot frames

A robot foot or leg can be represented as a chain of kinematic, which requires proper convention of frames for analysis. Furthermore, it is important to consider the components that comprise a single leg in order to establish mathematical relationships between its joints. In this thesis, each leg of the quadruped is comprised of 3 links: calf, thigh and hip, and features 3 joints that connects the 3 links: horizontally structured joint at the hip, the vertically structured joint at the hip, and the joint at the knee. Figure 3-3 illustrates the important components of a quadruped robot leg.

In a leg of the robot, the foot serves as the end effector, and it is crucial to note this fact, especially when identifying the starting point of the kinematic chain, which should be the joint of the horizontal hip.

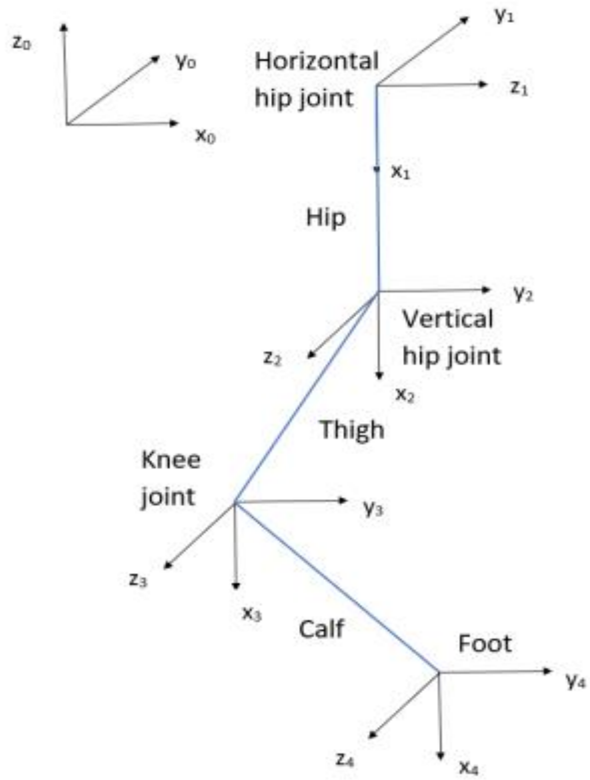

Figure 3-3 The Robot leg frames

There are 3 DOF for each leg, the joint of horizontal hip makes the adduction and abduction movements. The vertical hip joint enables extension and flexion of the thigh movements, whereas the knee joint allows for extension and flexion of the calf movements.

Between different frames, there are two types of movements: translation and rotation. The transformation matrix is a useful tool for establishing the mathematical relationship between these frames. To generate a transformation matrix that accounts for all the translations and rotations in each frame of the robot, a convention that encompasses both translations and rotations in every frame is essential.

The α angle determines the rotation around the x-axis, the φ angle determines the rotation around the y-axis, and the θ angle determines the rotation around the z-axis.

The rotation matrix of the robot with respect to the α angle is:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{3.1}$$

The matrix for the rotational transformation of the robot with respect to the φ angle is:

$$R_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \tag{3.2}$$

And lastly the θ angle rotation matrix is:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Regarding the translation matrices, the written symbol a is use to represent translation along the x-axis, the written symbol c is use to represent translation along the y-axis, and the written symbol d is use to represent translation along the z-axis.

The translation for the distance represented by the letter a can be defined as:

$$p_x = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

The translation for the distance represented by the letter c can be defined as:

$$p_y = \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix} \tag{3.5}$$

The translation for the distance represented by the letter d can be defined as:

$$p_z = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \tag{3.6}$$

Now, it is allowable create a matrix of homogeneous transformation by combining the rotation and translation matrices along each axis. This process generates a matrix of transformation for the x axis, which incorporates the angle α and distance represented by the letter a, a matrix of transformation for the y axis, which incorporates the angle φ and distance represented by the letter c, and a matrix of transformation for the z axis, which incorporates the angle θ and distance represented by the letter d. These matrices are respectively defined in Equations (3.7), (3.8), and (3.9).

$$T_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

$$T_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) & 0 \\ 0 & 1 & 0 & c \\ -\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

$$T_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

To obtain a transformation matrix that incorporates all the translations and rotations involved within every frame of the system, the three transformations must be multiplied as follows:

$$T_{i-1}^{i} = T_x(\alpha) T_y(\phi) T_z(\theta) = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix} \tag{3.10}$$

The elements of the developed transformation matrix are stated in Appendix A.

3.2.1.1 Forward Kinematics

Forward Kinematics determines the mathematical relationship between joint angles parameters and the end effector position. It can be derived using Denavit-Harteberg(D-H) convention by defining the link coordinate frames and then obtaining the relationship between the link and joint parameters. These parameters used in this convention are:

- The joint angle θ_i represents the amount of rotation in radians from the x_{i-1} axis to the x_i axis around the x_{i-1} axis.
- The joint distance d_i is the distance measured along the x_{i-1} axis from the coordinate system $(i-1)$ origin to the x_{i-1} axis and the x_i axis intersection.
- The link length l_i is the distance measured along the x_i axis from the intersection of the x_{i-1} axis and the x_i axis to the i^{th} origin system coordinate.
- The α_i link twist angle represents the amount of a rotational motion in radians from the z_{i-1} axis to the z_i axis around the x_i axis.

Therefore, by utilizing the Denavit Hartenberg parameters and establishing the appropriate robot leg frames relationship, it is allowable to determine the location of the foot end or the end effector relative to the main body or frame 0. The coordinate system for frame A is used as the starting point, with Equation (3.8) applied using angle φ of 90° and a distance c of 0. It is worth noting that the rotation matrix needs to be adjusted since the end effector of a robot leg is typically located at the bottom of its base, contrary to a typical kinematic chain where it is at the top. Table 3-1 is then used to determine the coordinate systems for the remaining frames, following the Denavit Hartenberg convention to establish the coordinate system linkage for the robot leg.

Table 3-1 DH Parameters for a Quadruped Robot Leg *(Aranda, 2022)*

D-H Parameters of the robot Leg				
Joint	θ_i	d_i	α_i	a_i
A_1^2	θ_1	0	90^0	L_1
A_2^3	θ_2	0	0^0	L_2
A_3^4	θ_3	0	0^0	L_3

The frames of a single leg are defined with respect to the robot frame 0, which implies to the joint of horizontal hip, as illustrated in Figure 3-4 below.

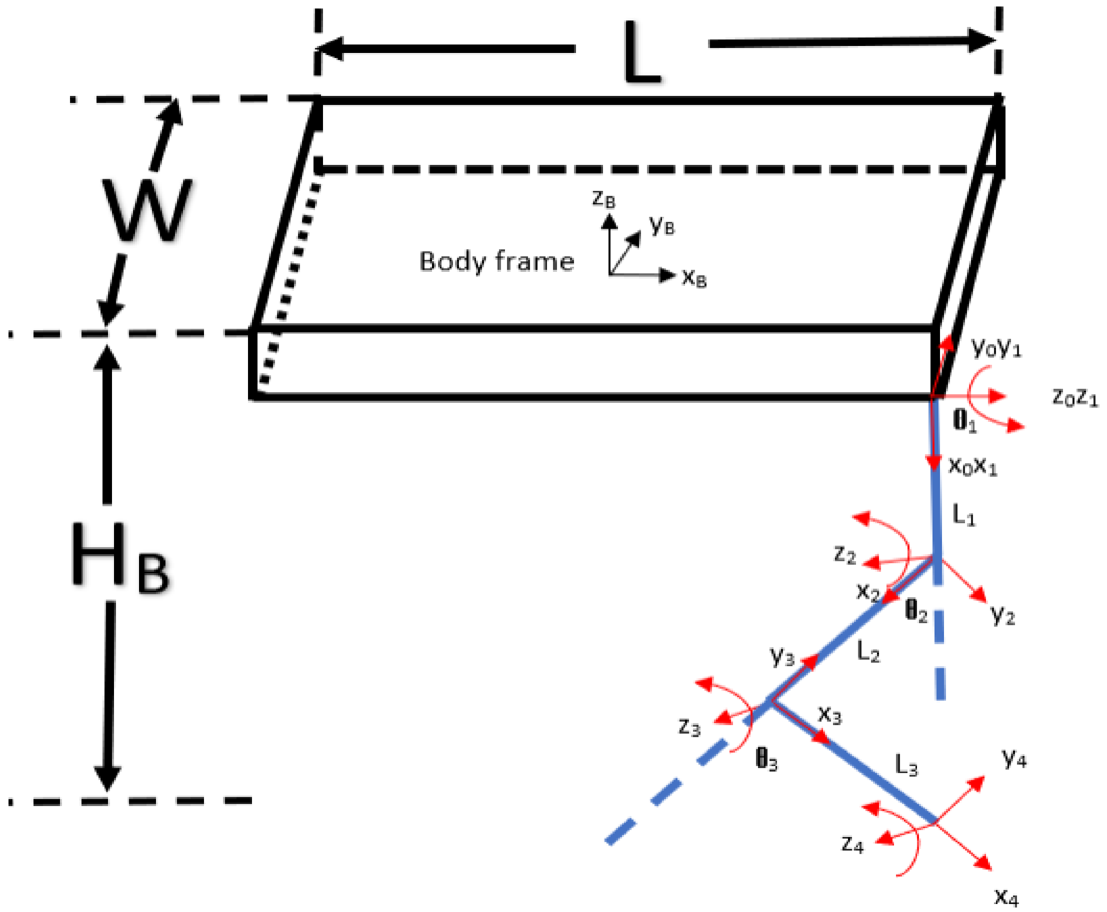

Figure 3-4 Single leg frames

Getting DH parameters, each transformation matrix can be obtained using Equations as follows:

$$A_0^1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

$$A_1^2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

$$A_2^3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

$$A_3^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

To obtain a matrix that represents the composition of the 4 transformation matrices from the frame of the foot to the base frame, they can be multiplied together using matrix multiplication, resulting in:

$$T_0^4 = A_0^1 A_1^2 A_2^3 A_3^4 = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{3.15}$$

Where:

$$R = \begin{bmatrix} \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 \\ \cos(\theta_2+\theta_3)\sin(\theta_1) & -\sin(\theta_2+\theta_3)\sin(\theta_1) & -\cos(\theta_1) \\ -\cos(\theta_2+\theta_3)\cos(\theta_1) & \sin(\theta_2+\theta_3)\cos(\theta_1) & -\sin(\theta_1) \end{bmatrix} \tag{3.16}$$

$$P = \begin{bmatrix} P_x \\ P_y \\ p_z \end{bmatrix} = \begin{bmatrix} L_3\sin(\theta_2+\theta_3) + L_2\sin(\theta_2) \\ \sin(\theta_1)(L_1 + L_3\cos(\theta_2+\theta_3) + L_2\cos(\theta_2)) \\ -\cos(\theta_1)(L_1 + L_3\cos(\theta_2+\theta_3) + L_2\cos(\theta_2)) \end{bmatrix} \tag{3.17}$$

The resulting P vector represents the forward kinematics and describes the relationship between the end effector and frame 0. This matrix can also be utilized to derive the leg dynamic model.

3.2.1.2 Inverse kinematics

Inverse Kinematics is used to determine the joint angles of the links at the known position of the quadruped robot. In contrast to forward kinematics, inverse kinematics analysis can be utilized to determine the end of leg effector trajectory using Cartesian coordinate system to obtain angles produced. To perform inverse kinematics, the position of the end-effector with respect to frame 0 is required and is represented by the parameters P_x, P_y, and P_z. Furthermore, the length of the calf (L_3), thigh (L_2) and hip (L_1), along with the length (L), width (W), and height (H_B) of the body are also necessary.

The purpose of utilizing the inverse kinematics algorithm is to calculate the angles of the leg that are used to track the trajectory reference of the foot. After some important algebraic mathematical operations are performed, the inverse kinematics is also obtained as follows:

$$\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{P_y + \frac{W}{2}}{P_z + \frac{H_B}{2}}\right) \\ -\varepsilon + \tan^{-1}\left(\dfrac{P_x - \frac{L}{2}}{\sqrt{\left(P_y + \frac{W}{2}\right)^2 + \left(P_z + \frac{H_B}{2}\right)^2} - L_1}\right) \\ \dfrac{L_2 + L_3}{L_3}\varepsilon \end{bmatrix} \tag{3.18}$$

Where:

$$\varepsilon = \cos^{-1}\left(\frac{L_2^2 + \left(P_x - \frac{L}{2}\right)^2 + \left(P_y + \frac{W}{2} - L_1 \sin\theta_1\right)^2 + \left(P_z + \frac{H_B}{2} + L_1 \sin\theta_1\right)^2 - L_3^2}{2L_2\sqrt{\left(P_x - \frac{L}{2}\right)^2 + \left(P_y + \frac{W}{2} - L_1 \sin\theta_1\right)^2 + \left(P_z + \frac{H_B}{2} + L_1 \sin\theta_1\right)^2}}\right) \tag{3.19}$$

Equations (3.16) and (3.18) provide the forward and inverse kinematics expressions, respectively. To establish the connection between the end effector position and the angles of leg rotation, the position of the end effector with respect to each angle is partially derived. This resulting expression, known as the contact Jacobian, could be determined as:

$$J_c = \begin{bmatrix} \dfrac{\partial x}{\partial \theta_1} & \dfrac{\partial x}{\partial \theta_2} & \dfrac{\partial x}{\partial \theta_3} \\ \dfrac{\partial y}{\partial \theta_1} & \dfrac{\partial y}{\partial \theta_2} & \dfrac{\partial y}{\partial \theta_3} \\ \dfrac{\partial z}{\partial \theta_1} & \dfrac{\partial z}{\partial \theta_2} & \dfrac{\partial z}{\partial \theta_3} \end{bmatrix} \tag{3.20}$$

Then, it yields:

$$J_c = \begin{bmatrix} 0 & L_2c_2 + L_3c_{23} & L_3c_{23} \\ L_1c_1 + L_2c_1c_2 + L_3c_1c_{23} & -L_2s_1s_2 - L_3s_1s_{23} & -L_3s_1s_{23} \\ L_1s_1 + L_2s_1c_2 + L_3s_1c_{23} & L_1c_1s_2 - L_3c_1s_{23} & L_3c_1s_{23} \end{bmatrix} \quad (3.21)$$

In Equation (3.21) above, s_i and ci represent sin (θ_i) and cos (θ_i), respectively, whereas s_{ij} and c_{ij} represent sin (θ_i+θ_j) and cos (θ_i+θ_j), respectively. The Jacobian matrix is utilized to evaluate the behavior of the controllers during experiments and ensure proper functionality.

3.2.1.3 The Legs Phases and Force Distribution

To effectively regulate the quadruped robot's legs locomotion, it is crucial to account for the two distinct phases they undergo during locomotion: the swing phase and the stance (support) phase. Coordination between these phases is necessary for proper whole body locomotion. The swing phase is featured by a lack of contact between the ground and the legs, resulting in the absence of contact force or ground reaction force. In contrast, during the stance phase, a contact force is present. Understanding the differences between these phases is essential when applying various controllers to ensure proper control of each phase and ultimately the entire locomotion process.

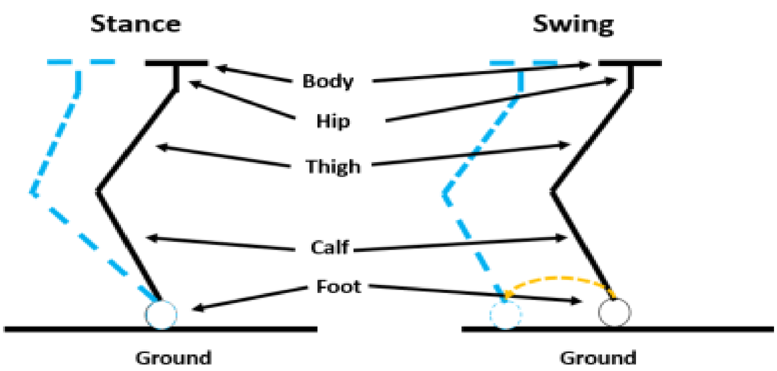

Figure 3-5 Stance and swing phases of quadruped legs locomotion

The blue dashed line in Figure 3-5 represents the swing and the stance final position at the end, respectively. Additionally, the dashed yellow line illustrates the trajectory of the foot during the swing phase of the leg motion. It can be observed that during the stance phase, the foot remains stationary while the main body moves. In contrast, during the swing phase, the body and the foot transition from their initial positions at the starting of this phase.

Maintaining stability when the quadruped robot is stationary requires proper and correct distribution of force in each leg that makes a contact with the ground. During the phase of support or stance, the leg determines the appropriate contact force amount to ensure the overall stability of the quadruped robot. According to (Xianpeng Zhang, 2015) the force acting on each foot during contact with the ground can be broken down into three different components, denoted as n_c, listed below:

$$\sum F_x = \sum_{t=1}^{n_c} Fx_i \tag{3.22}$$

$$\sum F_y = \sum_{t=1}^{n_c} Fy_i \tag{3.23}$$

$$\sum F_z = \sum_{t=1}^{n_c} Fz_i \tag{3.24}$$

Where:

$$f_i = \begin{bmatrix} Fx_i & Fy_i & Fz_i \end{bmatrix}^T$$

When the quadruped is in contact with the ground through its 4 legs, 4 forces are generated, as depicted in Figure 3-8, each of which has three vector components.

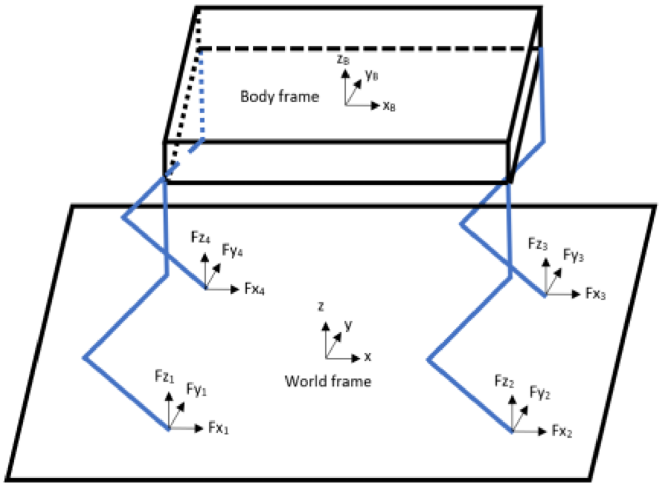

Figure 3-6 Force distribution of the legs of the quadruped

3.2.2 Dynamics Model of the Single Leg in Swing Phase

To calculate joint torques while the leg is swinging, it is necessary to obtain a model. Since reaction forces making contact are absent while the leg is swinging, the dynamic model can be defined as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.25}$$

In Equation (3.25), the variable $M(q) \in R^{3X3}$ implies the torque generated by the robot links inertia tensors, while $C(q, \dot{q}) \in R^{3X3}$ is a square matrix that implies the torque resulting from the forces of Coriolis and centrifugal. Additionally, $G(q) \in R^{3X1}$ denotes the torque generated by the gravity force, and $\tau \in R^{3X1}$ represents the robotic leg torques of the joint motors. Each leg of the robot has 3 degrees of freedom, and the joint motors control the abduction/adduction movements of the leg from the joint of horizontal hip, represented by $\theta 1$. The joint of vertical hip motors and knee motors control the thigh and calf flexion and extension respectively, denoted by θ_2 and θ_3. These angles can be represented as a vector as follows:

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \tag{3.26}$$

Their respective velocities can also be represented as:

$$\dot{q} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{3.27}$$

The dynamic model matrices can be calculated by first determining the center of mass (COM) position of the three links relative to frame zero belonging to the joint of horizontal hip. The transformation matrices derived previously can be utilized to determine the COM of the links. For instance, to obtain the COM position of the hip link, the transformation matrices $A_0^1 A_1^2$ can be multiplied, and the 1st, 2nd, and 3rd components of the 4th column can be extracted and divided by two, as demonstrated in Equation (3.28):

$$r1c = \begin{bmatrix} 0 \\ \frac{L1\sin\theta_1}{2} \\ \frac{-L1\cos\theta_1}{2} \end{bmatrix} \tag{3.28}$$

The center of mass (COM) coordinates for the thigh can be obtained in a similar way to the previous case by multiplying the transformation matrices $A_0^1 A_1^2 A_2^3$ and extracting the 1st three rows of the 4th produced matrix column. The COM thigh position relative to frame 0 can be determined by adding the hip full length plus the terms consisting of L_2 divided by two, resulting in Equation (3.29):

$$r2c = \begin{bmatrix} \frac{L_2\sin\theta_2}{2} \\ L_1\sin\theta_1 + \frac{L_2\cos\theta_2\sin\theta_1}{2} \\ -L_1\cos\theta_1 - \frac{L_2\cos\theta_1\cos\theta_2}{2} \end{bmatrix} \tag{3.29}$$

To determine the COM coordinates of the calf, the matrices $A_0^1 A_1^2 A_2^3 A_3^4$ can be multiplied, and only the first 3 rows of the resulting 4th column matrix should be considered. It should be noted that for the calf COM, the hip full length and the thigh full length together with the terms consisting of L_3 divided by two are taken into account, resulting in the COM position of the calf relative to frame 0. This can be expressed as Equation (3.30):

$$r3c = \begin{bmatrix} \frac{L_3\sin(\theta_2+\theta_3)}{2} + L_2\sin\theta_2 \\ \frac{\sin\theta_1(2L_1 + L_3\cos(\theta_2+\theta_3) + 2L_2\cos\theta_2)}{2} \\ -\frac{\cos\theta_1(2L_1 + L_3\cos(\theta_2+\theta_3) + 2L_2\cos\theta_2)}{2} \end{bmatrix} \tag{3.30}$$

To determine the center of mass (COM) links linear velocities relative to frame 0, the COM position partial derivative with respect to every angle of vector component q could be taken, as shown in Equation (3.31):

$$drn = \frac{\partial rc}{\partial q} \tag{3.31}$$

Once the velocities and positions of the links center of mass (COM) in the system have been determined, an Euler Lagrange method can be applied to calculate the potential and kinematic energy. The Lagrangian may be obtained by subtracting the potential energy of the system from the sum of the kinematic energy, as shown in Equation (3.32):

$$L = T - U \tag{3.32}$$

When calculating the kinematic energy, both the translational and rotational contributions of each link are taken into account, as depicted in Equation (3.33):

$$T = \frac{1}{2} m_i dr_n^T + \frac{1}{2} w_i^T I_\Delta w_i \tag{3.33}$$

It should be noted that the robot links are treated as circular cross-sectioned rigid bars and their I_Δ inertia tensors are considered, as shown in Equation (3.34):

$$I_\Delta = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} \frac{mL^2}{12} & 0 & 0 \\ 0 & \frac{mL^2}{12} & 0 \\ 0 & 0 & \frac{mL^2}{12} \end{bmatrix} \tag{3.34}$$

Each link makes a contribution to the overall kinematic energy of the system. Equation (3.33) can be used to determine the contribution of each link kinematic energy. Once all 3 links kinematic energy has been calculated, the system total kinematic energy can be obtained using Equation (3.35):

$$T = T_1 + T_2 + T_3 \tag{3.35}$$

The T matrix developed, along with the kinematic energy contribution of the three links, can be found in Appendix B.

The next step is to determine the contribution of each link to the potential energy, which can be obtained using Equation (3.36):

$$U = -m_i g^T r_n \tag{3.36}$$

By considering all the 3 links contributions, it is possible to calculate the entire leg potential energy by summing them up using Equation (3.37):

$$U = U_1 + U_2 + U_3 \tag{3.37}$$

The matrix U that has been derived can be found in Appendix C.

The dynamic model matrices elements presented in Equation (3.25) are determined based on the contributions of both the kinematic and potential energy. The M matrix consists of the leg links inertia tensor terms, which can be obtained by taking kinematic energy derivative, just like illustrated in Equation (3.38):

$$M(q) = \frac{\partial}{\partial dq}\left(\frac{\partial T}{\partial dq}\right) \tag{3.38}$$

The M matrix that has been derived can be found in Appendix D, which includes its elements.

The subsequent matrix to be computed is the C matrix, which accounts for the torques resulting from the Centrifugal and Coriolis terms. This can be determined using Equation (3.39) as shown it:

$$C(q,\dot{q}) = \frac{1}{2}(\frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k}) \tag{3.39}$$

The C matrix that has been derived can be found in Appendix E, which includes its elements.

To compute the torques resulting from gravitational forces, the matrix G can be derived by taking the derivative of the total potential energy with respect to the vector q each angle, as shown in Equation (3.40):

$$G(q) = \frac{\partial U}{\partial q_i} \tag{3.40}$$

The G matrix that has been derived can be found in Appendix F, which includes its elements.

3.2.3 Whole Body Dynamic Model

3.2.3.1 Lumped Mass Model

In order to have a comprehensive dynamic model for the entire quadruped during the swing phase, it is necessary to consider newton 2nd law to calculate the ground forces responses from legs in touch with the ground. The ground forces reaction can be determined using the lumped mass model, as described in Equations (3.41) and (3.42):

$$m\ddot{p} = \sum_{i=1}^{n_c} f_i - c_g \tag{3.41}$$

$$\frac{d}{dt}(I_R w) = \sum_{i=1}^{n_c} r_i \times f_i \tag{3.42}$$

Where:

$p \in R^3$: Position vector relative to the global frame.

$\ddot{p} \in R^3$: Acceleration vector relative to the world frame.

n_c: contacted legs count.

$c_g \in R^3$: Gravitational acceleration vector relative to the world frame.

$f_i \in R^3$: Reaction force vector relative to the world frame.

$IR \in R^{3\times3}$: Rotational inertia tensor.

m: Total mass of the robot.

$r_i \in R^3$: Position vector of the i^{th} contact point relative to the robot's center of mass.

$w \in R^3$: Body angular velocity of the robot.

Assuming that $w \times (I_w)$ is negligible for small angular velocities of bodies, Equation (3.42) can be approximated to yield Equation (3.43):

$$\frac{d}{dt}(I_R w) = I_R \dot{w} + w \times (I_R w) = I_R \dot{w} \tag{3.43}$$

Based on the aforementioned assumption, the angular acceleration and velocity of the center of mass be calculated using Equations (3.44) and (3.45):

$$\ddot{p} = \frac{\sum_{i=1}^{n_c} f_i - c_g}{m} \tag{3.44}$$

$$\dot{w} = I_R^{-1} \sum_{i=1}^{n_c} r_i \times f_i \tag{3.45}$$

The reaction forces f_i of each leg in contact with the ground are subsequently incorporated into Multi-Body Dynamics model to determine each leg joint torques.

3.2.3.2 Multi-Body Dynamic Model

Forces of reaction f_i obtained from model of the lumped mass for the legs in contact are utilized to calculate the torque commands for each leg joint using the Multi Body model. Acceleration $\ddot{p}$ vector from model of the lumped mass corresponds to the vector of $\ddot{q}$ from the multi-body dynamic model.

The multi-body dynamics model (Donghyun Kim, 2019) is described as equation (3.46)

$$S \begin{bmatrix} \tau_f \\ \tau_j \end{bmatrix} = A\ddot{q} + b + g - J_c^T f_i \tag{3.46}$$

Where:

$S = \begin{bmatrix} 0_{n_j x n_f} & I_{n_j} \end{bmatrix}$: The matrix of selection.

$q_f \;=\; [x, y, z, \phi, \theta, \varphi]^T$.

$q = \begin{bmatrix} q_j \\ q_f \end{bmatrix}$.

$q_j \in R_{nj}$: Joint coordinates of the actuated joints.

$q_f \in R^6$: Coordinates of the unactuated floating base.

n_j: Joint count.

$\ddot{q} \in R^3$: The vector that denotes the robot's acceleration with respect to the global frame, and joint accelerations vector relative to the global frame.

$b \in R^3$: Force due to Coriolis effect.

$A \in R^{3\times3}$: Overall mass matrix.

$g \in R^3$: Force of gravity.

$J_c \in R^{3\times3}$: Jacobian matrix for contact.

$\tau_f \in R3$: Torque at the floating base.

$\tau_j \in R3$: Torque at the joint.

$f_i \in R^3$: Force reaction from the lumped mass model.

3.2.4 Orientation of the COM and Angular Velocity

The robot's center of mass orientation can be described by Euler angles $\Theta = [\phi \quad \theta \quad \varphi]$, where ϕ represents roll, θ represents pitch, and φ represents yaw. Figure 3-7 provides a visual representation of the center of mass along with corresponding angles of rotation. These angles are important in constructing matrix of rotation that transforms the body into the world coordinates.

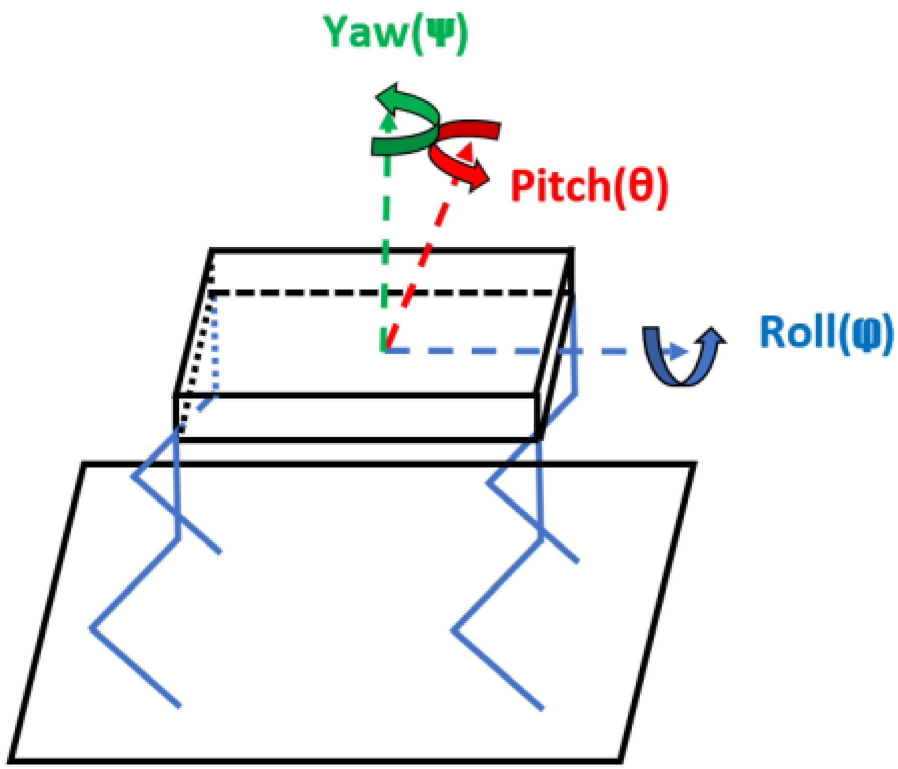

Figure 3-7 COM orientation

Using the matrices mentioned above, a rotation matrix can be established for each axis and angle that the quadruped rotates. The matrix of rotation for the angle ϕ can be defined by Equation (3.47):

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.47)$$

The matrix of rotation for the angle θ of the robot can be obtained using Equation (3.48):

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.48)$$

The matrix of rotation for the angle φ can be expressed using Equation (3.49):

$$R_z(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.49)$$

Using the aforementioned matrices of rotation, the transformation matrix which converts from the body to the world coordinates can be defined as illustrated in Equation (3.50):

$$R = R_z(\varphi) R_y(\theta) R_x(\phi) \quad (3.50)$$

The center of mass angular velocity can be computed by taking the first derivative of its orientation. To obtain the center of mass angular velocity with respect to the global frame, the previously derived matrix of rotation must be transposed (R^T). This can be represented by Equation (3.51), where the R^T matrix is multiplied by the center of mass angular velocity.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \dfrac{\cos(\varphi)}{\cos(\theta)} & \dfrac{\sin(\varphi)}{\cos(\theta)} & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ \cos(\varphi)\tan(\theta) & \sin(\varphi)\tan(\theta) & 1 \end{bmatrix} w \quad (3.51)$$

In most practical scenarios, the pitch and roll angles (θ and ϕ) are typically small, resulting in the factors containing $\cos(\theta)$ and $\cos(\phi)$ being approximately equal to one, whereas $\sin(\theta)$ and $\sin(\phi)$ are approximately equal to zero. Additionally, since tangent function is the ratio of sine and cosine, factors consisting $\tan(\theta)$ and $\tan(\phi)$ could easily be equal to zero.

By incorporating all of aforementioned simplifications, center of mass angular velocity can be determined using Equation (3.52), which is represented in terms of the simplified pitch and roll angles and the rotation matrix.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} w \tag{3.52}$$

This is equivalent to Equation (3.53):

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = R_z\left(\varphi\right)^T w \tag{3.53}$$

As previously mentioned, the quadruped robot can be observed from both the world and body frames. To establish a relation the inertia tensor in reference to the world frame to that in the body frame, a rotation matrix can be utilized. This relationship is established by Equation (3.54).

$$\hat{I}_w = R_z(\varphi)_B I R_z(\varphi)^T \tag{3.54}$$

3.2.5 Simplified Robot Dynamics

By employing the model of lumped mass, the motion equations for the center of mass or angular velocity and orientation can be derived, allowing for the dynamic of quadruped robot to be obtained. The entire robot dynamics can be expressed in a state-space representation, as illustrated in Equation (3.55):

$$\frac{d}{dt}\begin{bmatrix} \hat{\Theta} \\ \hat{p} \\ \hat{w} \\ \dot{\hat{p}} \end{bmatrix} = \begin{bmatrix} 0_3 & 0_3 & R_z(\varphi)^T & 0_3 \\ 0_3 & 0_3 & 0_3 & 1_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{p} \\ \hat{w} \\ \dot{\hat{p}} \end{bmatrix} + \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ \hat{I}_w^{-1}[r_1]\times & \hat{I}_w^{-1}[r_2]\times & \hat{I}_w^{-1}[r_3]\times & \hat{I}_w^{-1}[r_4]\times \\ \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \end{bmatrix} \tag{3.55}$$

The comprehensive dynamic model, which includes all the matrices expanded of Equation (3.55), found in Appendix G.

The value of the term $\hat{I}_w^{-1}[r_1]\times w$ can be determined by calculating the inertia matrix w.r.t the world frame, as illustrated in Equation (3.56) below:

$$\hat{I}_w = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.56}$$

Once performing necessary multiplications of matrix, the matrix of inertia in the world frame can be represented as shown in Equation (3.57):

$$\hat{I}_w = \begin{bmatrix} I_{xx}\cos(\varphi)^2 + I_{yy}\sin(\varphi)^2 & I_{xx}\cos(\varphi)\sin(\varphi) - I_{yy}\cos(\varphi)\sin(\varphi) & 0 \\ I_{xx}\cos(\varphi)\sin(\varphi) - I_{yy}\cos(\varphi)\sin(\varphi) & I_{yy}\cos(\varphi)^2 + I_{xx}\sin(\varphi)^2 & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.57}$$

Obtaining the inverse matrix of transformation w.r.t the world frame leads to Equation (3.58):

$$\hat{I}_w^{-1} = \begin{bmatrix} I_{xx} - I_{xx}\sin(\varphi)^2 + I_{yy}\sin(\varphi)^2 & \dfrac{\sin(2\phi)(I_{xx} - I_{yy})}{2} & 0 \\ \dfrac{\sin(2\varphi)(I_{xx} - I_{yy})}{2} & I_{yy} + I_{xx}\sin(\varphi)^2 + I_{yy}\sin(\varphi)^2 & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.58}$$

Then, the forward stage involves computing skew foot position matrix for a specific foot position, say r_1, where its position vector is given by $r_1 = [a_1\ b_1\ c_1]$. The skew matrix can be obtained using Equation (3.59):

$$[r_1]\times = \begin{bmatrix} 0 & -c_1 & b_1 \\ c_1 & 0 & -a_1 \\ -b_1 & a_1 & 0 \end{bmatrix} \tag{3.59}$$

Upon performing the inverse inertia matrix multiplication with the skew matrix for foot position with respect to the global frame, the matrix $\hat{I}_w^{-1}[r_1]\times$ can be derived. The expression for this matrix can be found in Appendix H.

In order to get a state-space representation equation that is more convenient, a new state for gravity was introduced. This expanded state includes all its components and can be expressed as Equation (3.60):

$$\dot{x}(t) = A_c(\varphi)x(t) + B_c(r_1, r_2, r_3, r_4, \varphi)u(t) \tag{3.60}$$

$$\dot{x}(t) = \begin{bmatrix} 0_3 & 0_3 & R_z(\varphi)^T & 0_3 & 0 \\ 0_3 & 0_3 & 0_3 & 1_3 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ \hat{I}_w^{-1}[r_1]\times & \hat{I}_w^{-1}[r_2]\times & \hat{I}_w^{-1}[r_3]\times & \hat{I}_w^{-1}[r_4]\times \\ \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} \\ 0 & 0 & 0 & 0 \end{bmatrix} u(t) \tag{3.61}$$

Where,

$$Ac = \begin{bmatrix} 0_3 & 0_3 & R_z(\varphi)^T & 0_3 & 0 \\ 0_3 & 0_3 & 0_3 & 1_3 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Bc = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ \hat{I}_w^{-1}[r_1]\times & \hat{I}_w^{-1}[r_2]\times & \hat{I}_w^{-1}[r_3]\times & \hat{I}_w^{-1}[r_4]\times \\ \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this simplified dynamic model equation Ac $\in$ $R^{13\times13}$ and Bc $\in$ $R^{13\times12}$.

CHAPTER FOUR

4 CONTROLLER DESIGN

The controller design is bifurcated into two components, namely the swing leg controller and the trunk balance controller. The swing leg is regulated using a proportional-derivative (PD) controller, while the trunk balance is maintained using a non-linear model predictive controller (NMPC).

4.1 Design of Trunk Balance Controller

The trunk-balance control (TBC) scheme is implemented to guarantee that the quadruped robot maintains a stable gait. By defining the variables in the reference state, such as the orientation, height, and velocity of the trunk, the required ground force support for the robot is determined using Model Predictive Control (MPC). The force output at the foot is determined by combining the reaction force and applied force, and the Jacobian matrix of the leg is utilized to compute the output each joint torque of the legs. The optimization solution and prediction equation for MPC are obtained based on the state-space dynamics model. The TBC diagram is depicted in Figure 4-1.

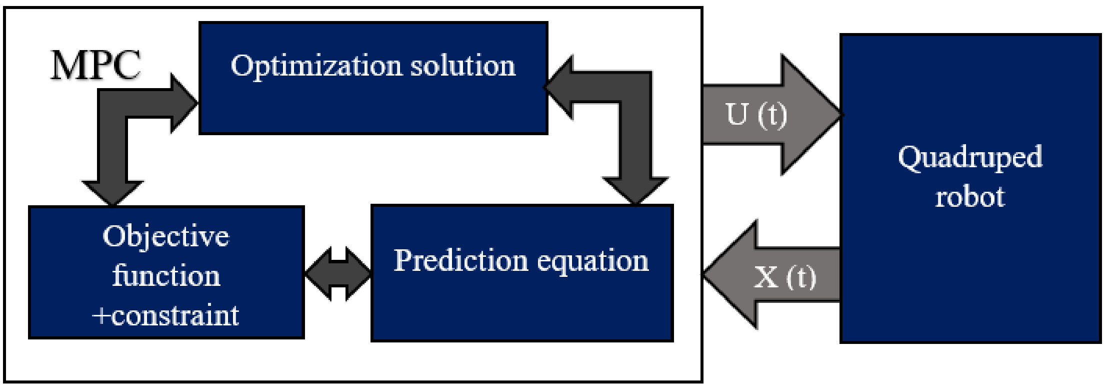

Figure 4-1 MPC algorithm of the Trunk balance controller

4.1.1 Prediction equation

As previously mentioned, the state space dynamic model of quadruped robots, which is well-suited for model prediction purposes, is given by:

$$\dot{x}(t)=\begin{bmatrix}0_3 & 0_3 & R_z(\varphi)^T & 0_3 & 0\\ 0_3 & 0_3 & 0_3 & 1_3 & 0\\ 0_3 & 0_3 & 0_3 & 0_3 & 0\\ 0_3 & 0_3 & 0_3 & 0_3 & 1\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}x(t)+\begin{bmatrix}0_3 & 0_3 & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3\\ \hat{I}_w^{-1}[r_1]\times & \hat{I}_w^{-1}[r_2]\times & \hat{I}_w^{-1}[r_3]\times & \hat{I}_w^{-1}[r_4]\times\\ \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m} & \frac{1_3}{m}\\ 0 & 0 & 0 & 0\end{bmatrix}u(t) \quad (4.1)$$

Therefore, having this model the discretization forward Euler process can be conducted as shown below which enable the model fit to the model predictive controller utilization.

$$x(k+1)=A_k x(k)+B_k u(k) \tag{4.2}$$

Where,

$$x=\begin{bmatrix}\Theta^T & p^T & w^T & \dot{p}^T & G\end{bmatrix},$$
$$u=\begin{bmatrix}f_1 & f_2 & f_3 & f_4\end{bmatrix},$$
$$A_k=(I+TA_c)=\begin{bmatrix}1_3 & 0_3 & T*R_z(\varphi)^T & 0_3 & 0\\ 0_3 & 1_3 & 0_3 & T*1_3 & 0\\ 0_3 & 0_3 & 1_3 & 0_3 & 0\\ 0_3 & 0_3 & 0_3 & 1_3 & T\\ 0 & 0 & 0 & 0 & 0\end{bmatrix},$$

$$B_k=TB=\begin{bmatrix}0_3 & 0_3 & 0_3 & 0_3\\ 0_3 & 0_3 & 0_3 & 0_3\\ \hat{I}_w^{-1}[r_1]\times T & \hat{I}_w^{-1}[r_2]\times T & \hat{I}_w^{-1}[r_3]\times T & \hat{I}_w^{-1}[r_4]\times T\\ \frac{1_3*T}{m} & \frac{1_3*T}{m} & \frac{1_3*T}{m} & \frac{1_3*T}{m}\\ 0 & 0 & 0 & 0\end{bmatrix},$$

T_s is sample time

Once dynamic model is discretized, it can be used for the prediction of the future system characteristic's over a finite prediction horizon. The optimization problem involves minimizing a cost function subjected to constraints for the system control inputs and state. The optimal control action is determined by solving the optimization problem at each step time using numerical optimization techniques. The initial control action is utilized to the system, and the algorithm is repeated at the next time step.

In subsequent h steps, state equation for the system can be represented as illustrated:

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \\
x_{k+2} &= Ax_{k+1} + Bu_{k+1} = A(Ax_k + Bu_k) + Bu_{k+1} \\
&= A^2 x_k + ABu_k + Bu_{k+1} \\
x_{k+3} &= A^3 x_k + A^2 Bu_k + ABu_{k+1} + Bu_{k+2} \\
&. \\
&. \\
&. \\
x_{k+h} &= A^h x_k + A^{h-1} Bu_k + ... + ABu_{k+1} + Bu_{k+h-1}
\end{aligned}
\tag{4.3}
$$

Finally, the state-space future trajectory equation of the system is given by::

$$\bar{X} = AX + BU \tag{4.4}$$

Where,

$$
\begin{aligned}
\bar{X} &= [x_{k+1} \quad x_{k+2} \quad ... \quad x_{k+h}]^T, \\
U &= [u_k \quad u_{k+1} \quad ... \quad u_{k+h-1}]^T, \\
A &= [A \quad A^2 \quad ... \quad A^h]^T, \\
B &= \begin{bmatrix} B & 0 & 0 & ... & 0 \\ AB & B & 0 & ... & 0 \\ A^2 B & AB & B & ... & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ A^{h-1}B & A^{h-2}B & A^{h-3}B & ... & B \end{bmatrix}
\end{aligned}
$$

4.1.1.1 Quadratic Optimization

The control law U in the aforementioned state equation is not known a priori, and can only be obtained through an optimization process. In the time domain of control, U is determined by defining a function of suitable objective, which is then used to generate an optimization solution. The objective function is illustrated as:

$$L(x,\ \mathrm{u}) \ = \ \min_{x,u} \sum_{i=1}^{Np} \left\| x(i+1) - xref\,(i+1) \right\|_Q^2 + \sum_{i=1}^{Nc-1} \left\| u(i) \right\|_R^2 \tag{4.5}$$

In order to allow the quadruped robot to effectively track the desired value or trajectory with minimal control input, the current input must be optimized. The objective function defined in Equation (4.5) consists of two terms: the first term assesses the system's ability for tracking the trajectory reference, while the 2nd term ensures that the changes of control quantity without any abrupt changes or disruptions. The matrices Q and R are diagonal weighting matrices, while N_p and N_c represent the prediction and control horizons, respectively. The objective function is designed to facilitate quick and smooth tracking of the desired trajectory by the robot.

In this study, the constraint limits of both the state and control variables are taken into account during the control process. The expression of both constraints quantity which the value will be given in the parameters list are:

$$St:\begin{cases} x_{i+1} = A_i x_i + B_i u_i, i = 0,...N_p - 1 \\ u_{\min} \le u_z \le u_{\max} \\ -\mu u_z \le u_x \le \mu u_z (\mu \text{ is cofficient of friction}) \\ -\mu u_z \le u_y \le \mu u_z \end{cases} \tag{4.6}$$

To obtain the optimization solution of the MPC at every step, the problem can be contructed as a quadratic programming problem, which can be expressed as follows:

$$\min_f(\frac{1}{2}U^T HU + R^T U) \tag{4.7}$$

Where,

$$\begin{aligned} &cmin \ \le \ CU \ \le \ cmax, \\ &H \ = \ 2\left(B^T QB \ + \ \mathrm{R}\right), \\ &V \ = \ 2B^T Q\left(x \ - \ xre\, f \ \right). \end{aligned} \tag{4.8}$$

Upon solving the aforementioned quadratic programming problem at each control cycle, a set of control inputs can be generated within the control-time domain:

$$U \ = \ \left[U_k, \ U_{k+1}, \ U_{k+2}, \ \cdots, \ U_{k+h-1} \ \right] \tag{4.9}$$

The Model Predictive Controller operates on the principle of applying the initial control input within the system control-time domain, followed by the computation of the output torque for each joint by leg jacobian matrix transposition. In each successive control cycle, the system utilizes the current state information to re-predict the state variables in the upcoming time domain, and then conducts a procedure of optimization to generate fresh set of manipulation inputs. This loop is repeated until the quadruped robot finishes the process of regulation.

Finally, to compare the designed controller optimization based PID controller is considered in the centroid trajectory. The control law that governs the PID controller is:

$$u(t) = Kp * e(t) + Ki * \int_0^t e^*(t)dt + Kd * \frac{de(t)}{dt} \tag{4.10}$$

Where, K_p is proportional gain, K_i is integral gain and K_d is derivative gain.

4.2 Design of Swing Leg Controller

The swing leg controller is designed for purpose of controlling the swing leg motion of the quadruped robot. The controller which is used for this control action is PD controller. The equation of control action for PD controller is as shown below:

$$u = K_p * e + K_d * \frac{de}{dt} \tag{4.11}$$

Where, e is error, de is change in error, K_p is proportional constant and K_d is derivative constant parameter.

In this work each legs of the robot consists of three DOF. The input of the joints is the torque which causes the motion. These torques are three since there are three joints which are motors in real time system usually servo motors to be controlled. Therefore, the error and change in error of the leg control can be obtained as follows:

$$\begin{cases} e = qdesired - q \\ de = \dot{q}desired - \dot{q} \end{cases} \tag{4.12}$$

Where,

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}, \qquad \dot{q} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}$$

The controller parameters K_p and K_d are also stated as follows:

$$\begin{cases} K_p = \begin{bmatrix} K_{p1} & 0 & 0 \\ 0 & K_{p2} & 0 \\ 0 & 0 & K_{p3} \end{bmatrix} \\ K_d = \begin{bmatrix} K_{d1} & 0 & 0 \\ 0 & K_{d2} & 0 \\ 0 & 0 & K_{d3} \end{bmatrix} \end{cases} \tag{4.13}$$

Generally, by including the dynamic model equation of the leg the control action $u = \tau = [\tau_1 \quad \tau_2 \quad \tau_3]$ can be obtained as follows:

$$\tau_i = K_p(qdesired - q)_i + K_d(\dot{q}desired - \dot{q})_i + \text{ additional torques}_i \tag{4.14}$$

4.3 Tuning Parameters of Controllers

Tuning of the controllers parameters is an important activity to meet objective of this work which is obtaining high trajectory tracking accuracy for the quadruped robot. These parameters can be tuned in two ways: manual and automatic tuning ways. In this thesis work the controllers parameters are tuned automatically with a metaheuristic optimization algorithm called particle swarm optimization algorithm.

4.3.1 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is an optimization technique inspired by nature and developed by Kennedy and Eberhart. It is a powerful and simple algorithm that has been successfully applied in various scientific and engineering fields. At the start, the PSO system comprises a population of solution values randomly chosen from a distribution. Each solution value is called a particle, and it has a randomly selected velocity. Additionally, each particle has a position known as its best position (P_{best}), and particles move while keeping track of their P_{best}. For each P_{best}, there is a corresponding fitness value, and the particle with the highest fitness value is referred to as the global best (G_{best}).

There are two important main equations in the metaheuristic algorithm of the PSO, position and velocity vectors respectively shown below in equations (4.15) and (4.16):

$$v_{p,q}(t+1) = wv_{p,q}(t) + r_1 c_1 (p_{p,q}(t) - x_{p,q}(t)) + r_2 c_2 (g_q(t) - x_{p,q}(t)) \tag{4.15}$$

$$x_{p,q}(t+1) = x_{p,q}(t) + v_{p,q}(t+1) \tag{4.16}$$

Here, r_1 and r_2 are two random vectors whose values range between 0 and 1. The parameters c_1 and c_2 represent learning parameters. The variable x represents the position, while the variable v represents the velocity.

4.3.1.1 PSO Algorithm

The PSO algorithm comprises the following steps:

1. Create an array of individuals with randomly assigned positions and velocities that satisfy the inequality constraints.
2. Check whether the equality constraints are satisfied and adjust the solution as necessary.
3. Evaluate the fitness function for each individual.
4. Compare the current fitness value with the individual's previous best value (p_{best}). If the current fitness value is lower, update p_{bestx} with the current coordinates (positions).
5. Determine the current global minimum fitness value among the current positions.
6. Compare the previous global minimum (g_{best}) with the current global minimum. If the current global minimum is better than g_{best}, update g_{bestx} with the current coordinates (positions).
7. Update the individual's position and velocity according to the PSO algorithm.
8. Repeat Steps 2-7 until the optimization is satisfied or the maximum number of iterations is reached.

The objective function considered for the particle swarm optimization algorithm is Integral of the Absolute Magnitude of the Error (IAE) is determined by equation for all controllers in this thesis:

$$IAE = \int_0^T |e(t)|\, dt \tag{4.17}$$

Table 4-1 PSO algorithm parameters

PSO parameters	Values
Number of iteration	50
Population size	30

The flow chart of PSO algorithm is illustrated as follows:

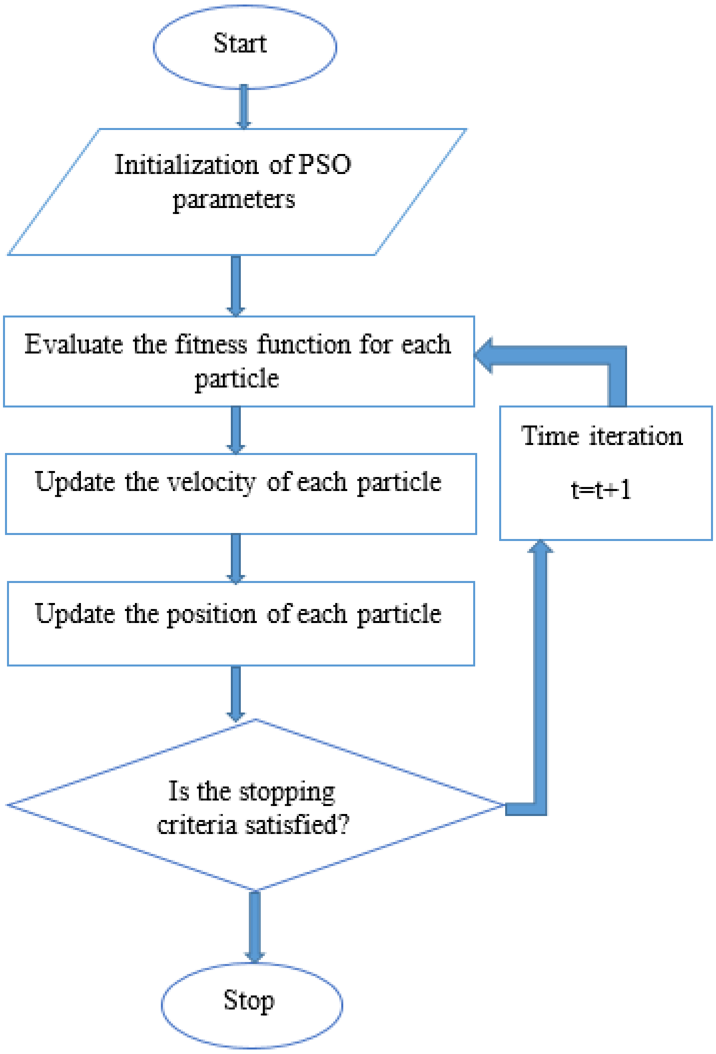

Figure 4-2 Flow chart of PSO algorithm

In this thesis work, the particle swarm MATLAB function from global optimization toolbox is used for tuning of both controllers' parameters.

CHAPTER FIVE

5 RESULT AND DISCUSSION

5.1 Simulation Results

The simulation of both the COM and swing leg dynamic model is carried out using MATLAB command line. In quadruped robot simulation, it is crucial for analyzing the locomotion arrangement of the legs to achieve a realistic imitation of animal gaits. To guarantee correct movement, every leg needs to go through two phases of movement: the stance and the swing phase. During the stance phase, the leg is touching the ground, and a reaction ground force is generated. In contrast, during the swing phase, no contact force is present as the leg is not touching the ground.

Considering two main phases of the quadruped legs locomotion, the torques can be computed using PD controller as shown below:

$$\tau_i = K_p(qdesired - q) - K_d(\dot{q}desired - \dot{q}) + additional\ torques_i \qquad (5.1)$$

It is noteworthy that while the leg is swinging, the torques are calculated using the PD controller described in Equation (5.1), whereas in the stance phase, extra torques are generated through the use of a Quadratic Programming (QP) solver by the MPC. The QP solver computes the contact forces of the robot legs. However, this thesis work concentrates solely on the swing control of leg locomotion. Therefore, additional torque is not taken into account for leg locomotion since it occurs during the stance phase.

The gait of diagonal trot is among the motions that are the most efficient for quadruped robots, and therefore, this work's controller takes an analytical approach to consider this gait. In this walking pattern, the 2-legs on the robot's trunk diagonal are in a same cycle, whereas other 2-legs on opposite diagonal are in the contrary cycle. To preserve dynamic balance throughout the motion, both the TBC and SLC are activated in a continuous manner for each leg.

Table 5-1 The robot physical parameters

Parameters	Symbols	Values	Units
Mass of the body	m	9	kg
Gravitational acceleration	g	9.81	m/s^2
Inertia of the body	I_{xx}	0.112	$kg.m^2$
Inertia of the body	I_{yy}	0.362	$kg.m^2$
Inertia of the body	I_{zz}	0.426	$kg.m^2$
Length of the body	L_{body}	0.6	m
Width of the body	W_{body}	0.3	m
Height of the body	H_{body}	0.47	m
Hip length	L_1	0.072	m
Thigh length	L_2	0.2	m
Calf length	L_3	0.2	m
mass of the links	(m_1,m_2,m_3)	(0.2264,0.2112,0.264)	kg
Sampling period	T_s	15	ms
Friction factor	μ	0.3	none

5.1.1 Simulation of COM Trajectory Tracking

In this thesis work, two centroid trajectory simulations have been done. These are:

- Linear trajectory and
- Circular trajectory.

In linear trajectory, the desired reference given is 0.3m so that the robot track it with fast convergence and minimized steady state error performance. From the initial point of the robot standing condition the system is simulated. In circular trajectory, the desired reference given is a circle of radius 0.3m so that the robot track it with fast convergence and minimized steady state error performance. From the given parameters in the above table the friction factor is required for the purpose of the limit constraints. So, the controller is designed with the assumption of foot friction reaction forces and the lower and upper limits of them is [-26, -26, -88] and [26, 26, 88] respectively

5.1.1.1 Linear Trajectory Tracking Simulation

As previously mentioned, two main frames are present in the quadruped namely the global frame and the local frame. This work focuses on the robot's locomotion within the world frame, enabling the camera to rotate and observe the robot from different perspectives. Linear trajectory tracking for the quadruped robot is done for COM position in the z-axis. The MPC tracking controller is designed and the advantages of high accuracy is obtained.

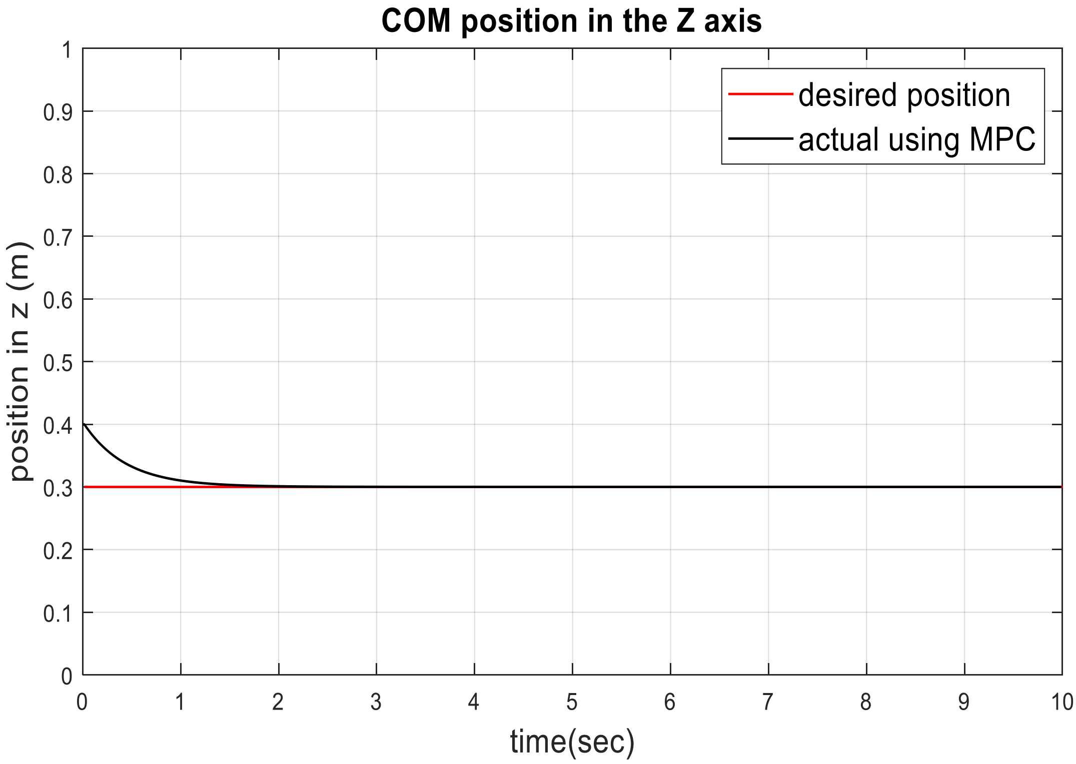

Figure 5-1 COM position linear Trajectory in the z axis

The MPC controller, as depicted in Figure 5-1, demonstrated a rapid and precise tracking ability with minimal initial tracking error. Additionally, the entire trajectory tracking process remained stable. The steady-state error for this trajectory is 9.7702e-05m or 0.0097702%. The preceding figure illustrates a simulation test comparing the actual z axis center of mass position with the desired center of mass position.

Even when the body experiences a random disturbance, such as a pushing input, it can still track the desired COM position and achieve stability rapidly, which is shown in Figure 5-2, where the quadruped robot experiences a disturbance.

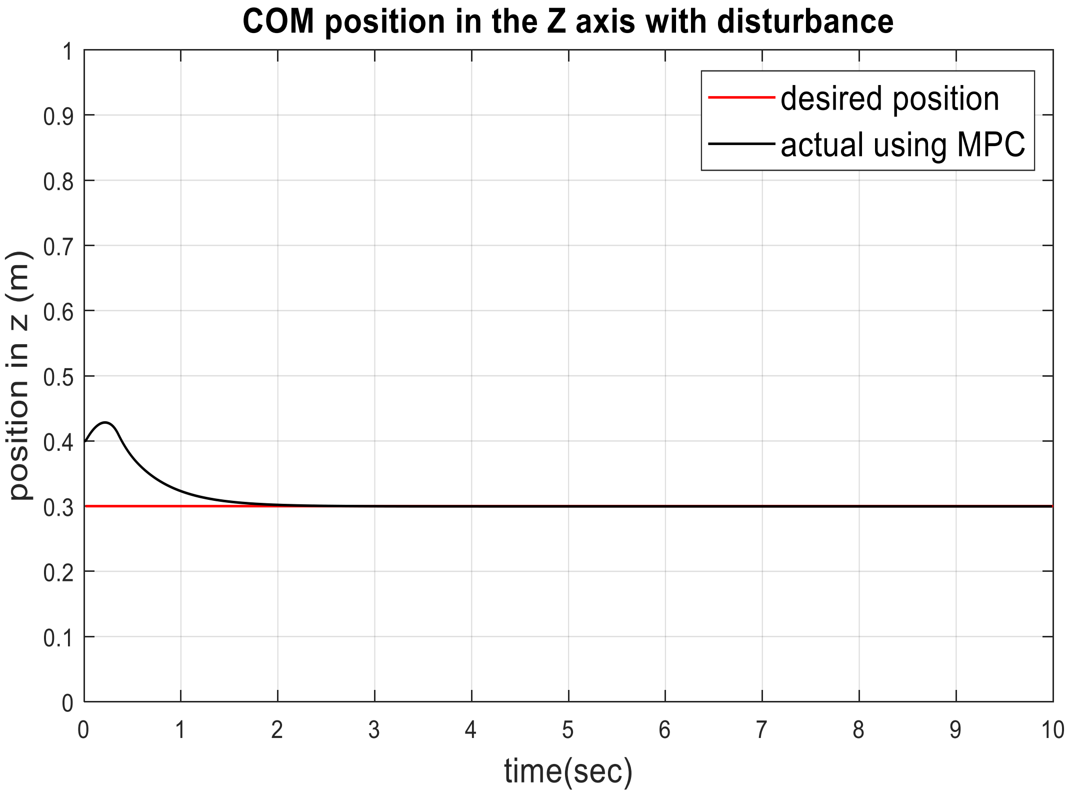

Figure 5-2 position of COM linear Trajectory in the z axis subjected to disturbance

As it can be seen from the figures above the system settles to steady state within less than 3 seconds when its subjected to disturbance and when its not subjected.

5.1.1.2 Circular Trajectory Tracking Simulation

The second COM simulation is tracking of the quadruped's robot circular path. A model predictive based control approach developed in this work can enhance the precision of the motion tracking along a trajectory while exhibiting excellent steady-state performance.

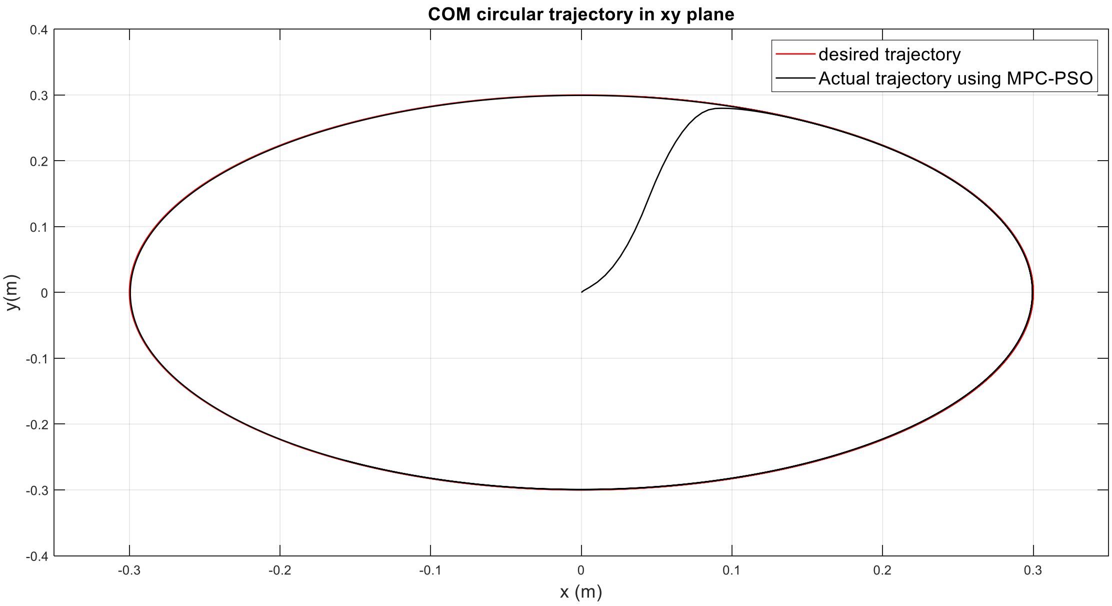

Figure 5-3 Centroid Circular path tracking of the quadruped robot's movement

Objective function minimized value called integral of the absolute error minimized is 0.0136m, which gives the optimal value of the parameters of MPC and some the parameters are adjusted by hand. The error in the x and y axis during simulation are as shown below:

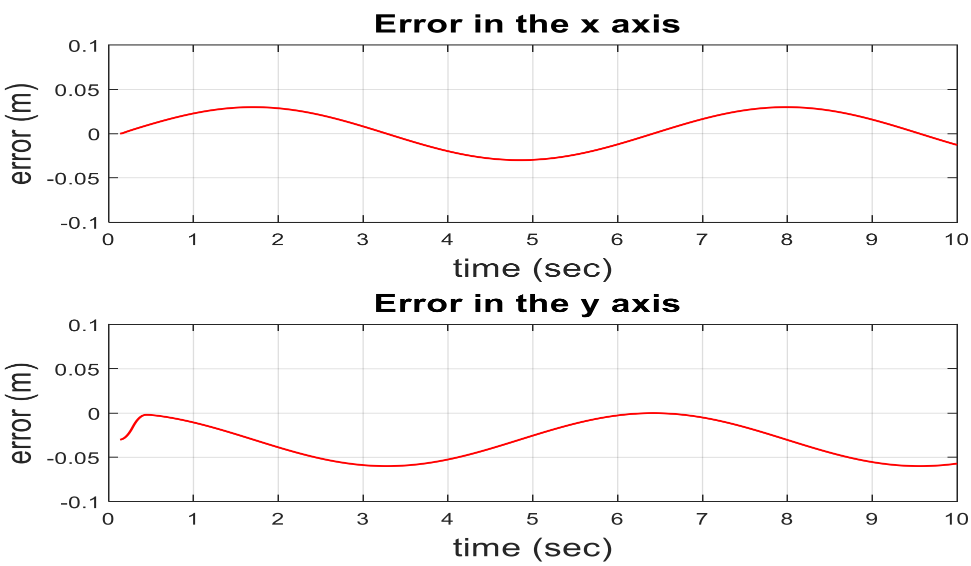

Figure 5-4 Errors in the x and y axis of the centroid trajectory with MPC controller

The designed controller which is model predictive controller with prediction horizon of 10 is applied to the system with the following gains as depicted in the Table5-4.

Table 5-2 Upper and lower parameters values of MPC weights

Parameter	Lower bound	Upper bound
Q(4,4)	280	300
Q(5,5)	270	300

Table 5-3 Weighting matrices of model predictive controller

Gain parameters applied to COM tracking MPC		
Trajectory	**Q**	**R**
Linear trajectory tracking case	diag([1, 1, 1, 28, 1, 20, 1, 1, 94, 1, 1, 1])	0.00000111*diag([ones(1,12)])
Circular trajectory tracking case	diag([100,100,100,285.709, 273.096, 260, 1, 1, 0.008, 1, 1,1])	0.000000222*diag([ones(1,12)])

The proposed and designed controller for the quadruped robot have also been compared with an optimization based PID controller. The PID controller is less accurate than MPC in tracking the given circular trajectory as shown in the figure below:

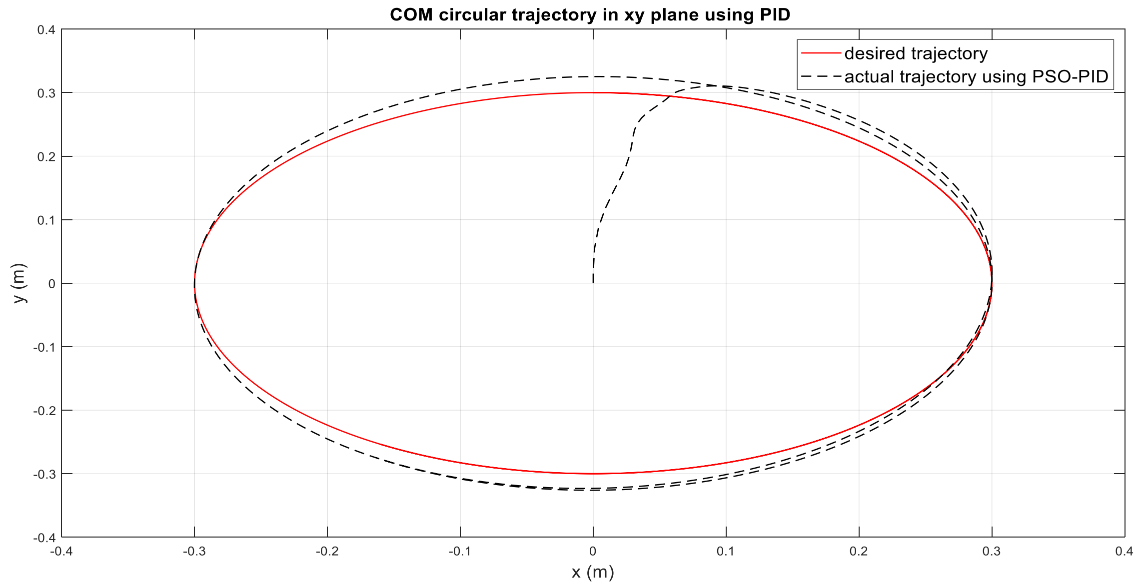

Figure 5-5 Centroid Circular trajectory tracking of the quadruped robot using PID

The error in the x and y axis during the simulation of the trajectory with PID controller are as shown below:

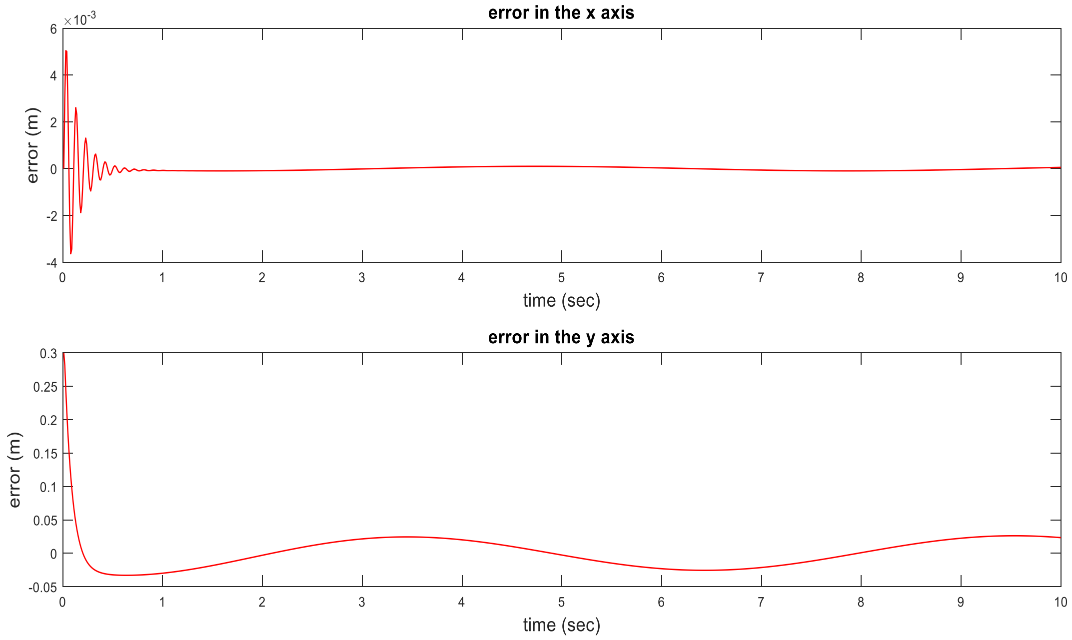

Figure 5-6 Errors in the x and y axis of the centroid trajectory with PID controller

The minimized value of the objective function called integral of the absolute error minimized is 0.2098m, which gives the optimal value of the parameters of PID and some of the parameters are adjusted by hand. The objective function diagram can also be seen as follows:

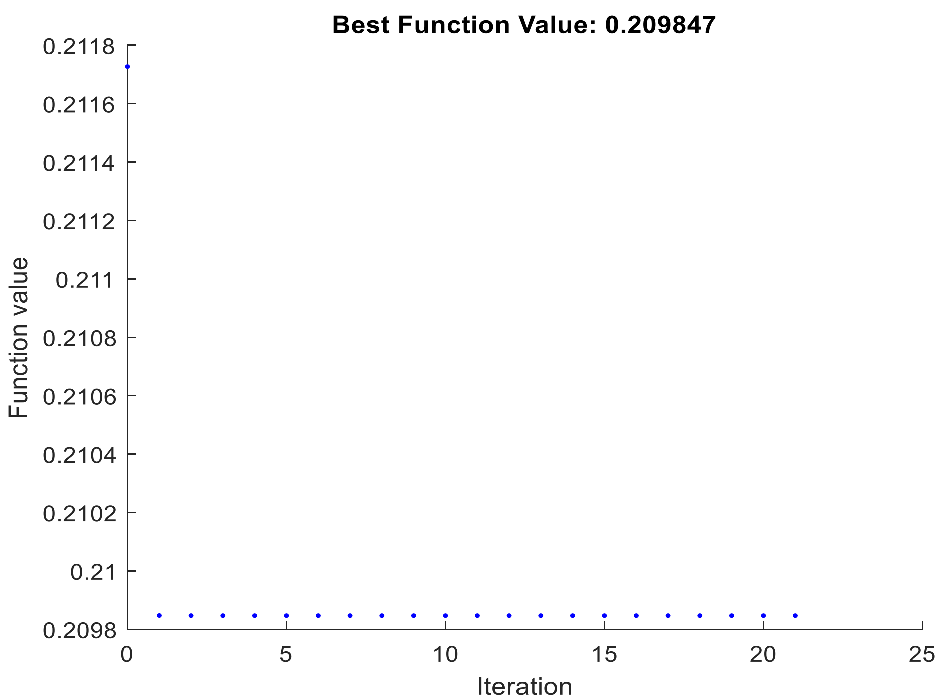

Figure 5-7 Objective function minimized

Table 5-4 Upper and lower bounds of PID parameter values

Parameter	Lower bound	Upper bound
Kp (4,4)	27550	27600
Kp (5,5)	100	110
Kd (4,4)	280	300
Kd (5,5)	100	110

Table 5-5 Gains of PID controller using PSO

PID gain parameters used in COM trajectory	
K_p	diag([0, 0, 0, 27600, 110, 0, 0, 0, 1000, 0, 0, 0])
K_i	Diag([0 0 0 100 100 0 0 0 0 0 0 0])
K_d	diag([0, 0.0, 0, 300, 110, 0, 0, 0, 100, 0, 0, 0])

5.2 The Swing Leg Controller Simulation

This thesis presented and designing of the controller for leg swing intended for evaluating swing phase of a single leg in the locomotion of the quadruped robot system. For the robotic leg simulation, the leg dynamic model with 3 DOF is used. Totally there are 12 degree of freedom which becomes 24 states with the corresponding velocities. In this work the simulation of single leg is considered which have 6 states with corresponding velocities.

By utilizing the proposed model for leg dynamics, the $\ddot{q}$ vector can be derived and utilized in control. Equation (5.2) provides the formula for the actual acceleration.

$$\ddot{q} = M(q)^{-1}(\tau - C(q,\dot{q}) - G(q)) \tag{5.2}$$

The real acceleration vector can be integrated to determine the actual velocity vector, $\dot{q}$, and the velocities vector can also be integrated to obtain the actual position vector, q.

The desired reference trajectories given for the leg dynamic model are shown below in Figure 5-8 verify the behavior of the angles so as to make robot leg move stably and follow the pattern like animals do. Differentiating the q_{ref} vector yields the reference velocities vector, $\dot{q}_{ref}$, while differentiating the reference velocity produces the reference accelerations vector, $\ddot{q}_{ref}$. Figure 5-8 displays the angle reference.

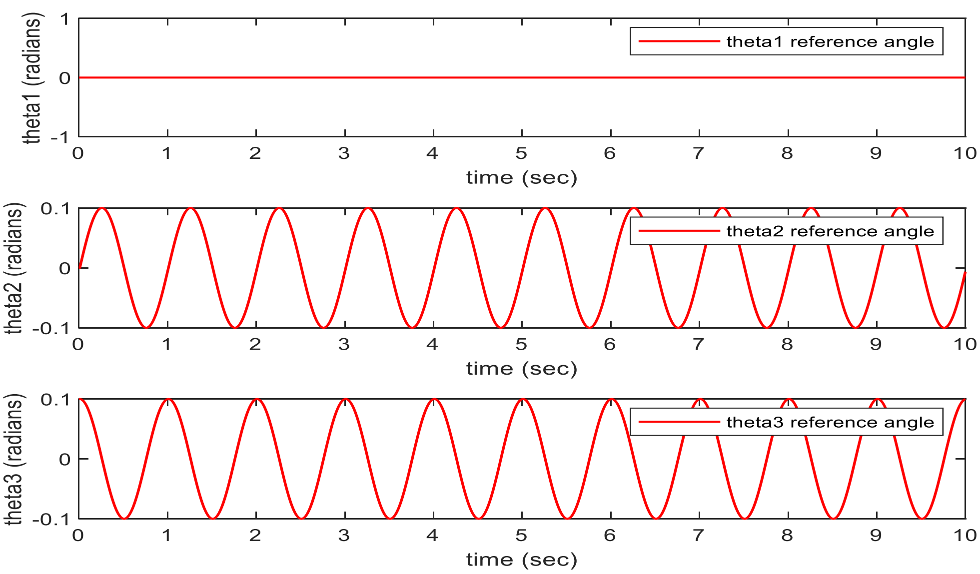

Figure 5-8 Reference angles

Figure 5-9 illustrates the derivation of the angle reference position and the subsequent acquisition of the reference velocities vector. The velocity reference employed in the simulation is also visible.

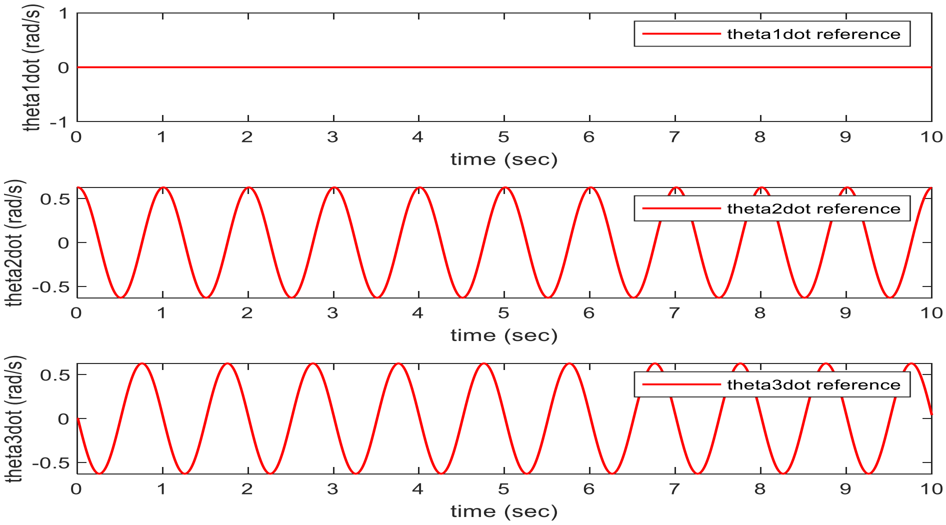

Figure 5-9 the reference of the velocities

The system is simulated for 10 seconds. Following the simulation, system angle graphs were generated, comparing the actual and reference angles during the simulation. The behavior of the horizontal hip joint is represented by the θ_1 angle, whose reference angle is consistently zero throughout the simulation. Figure 5-10 depicts how the actual system angle accurately and rapidly tracks the provided reference angle. The θ_2 angle represents the joint vertical hip behavior, following the reference desired trajectory, while the θ_3 angle represents the knee joint behavior, also following the reference desired trajectory.

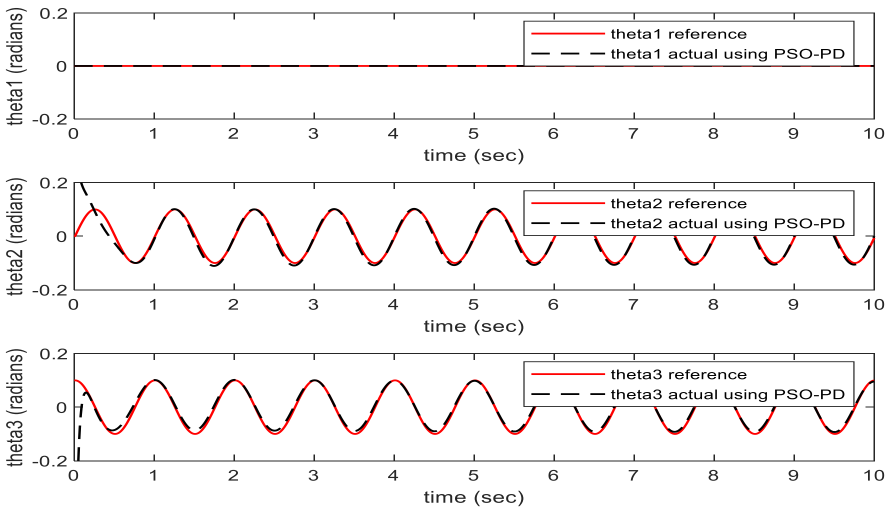

Figure 5-10 Actual angles versus reference angles

Therefore, the particle swarm optimization based PD controller have best performance for all the three angles of the leg which are determinants for the proper movement of the leg. During simulation, the horizontal hip joint angular velocity was represented by $\dot{\theta}_1$, it is reference value is zero for the whole simulation, as it can be seen in Figure 5-11 the actual or real value of the system tracked the given reference accurately and gave a fast response. The angular velocity $\dot{\theta}_2$ observed during simulation corresponds to the vertical hip joint angular velocity, as it follows the reference desired trajectory. Likewise, the angular velocity $\dot{\theta}_3$ observed during simulation, it denotes the rate of change of knee joint angle, as it follows the intended reference path, just like illustrated in the figure below. Therefore, the particle swarm optimization based PD controller have best performance for all the three angular velocities of the leg which also contributes for the proper movement of the leg.

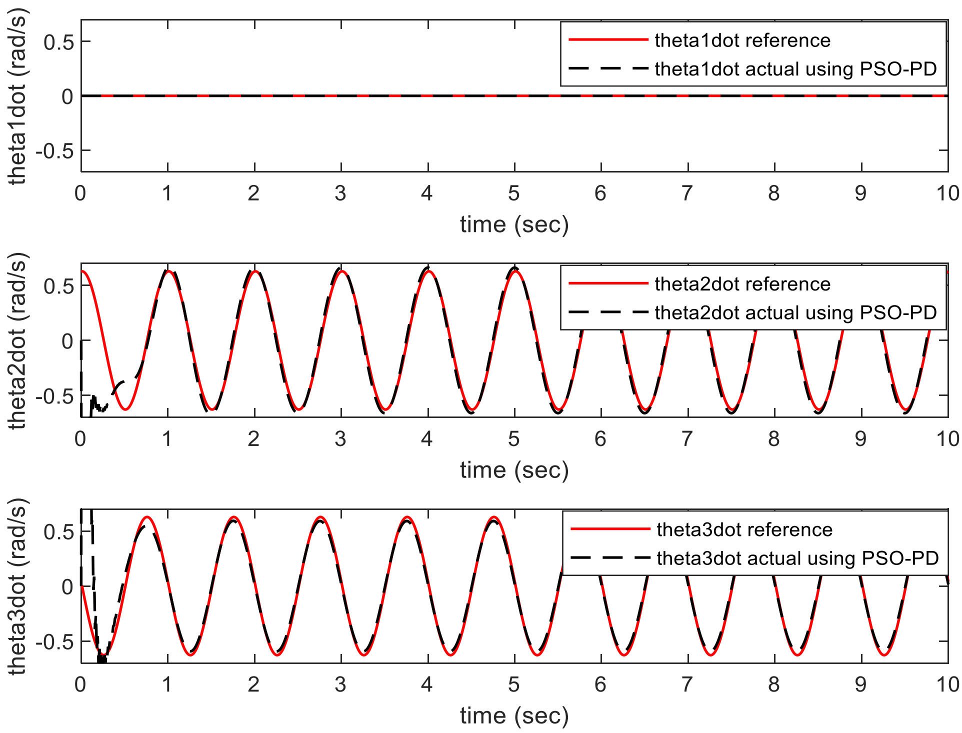

Figure 5-11 Actual angular velocities versus reference velocities

Once the control was implemented and the system adequately tracked the reference positions for every joint, corresponding torques observed as depicted below:

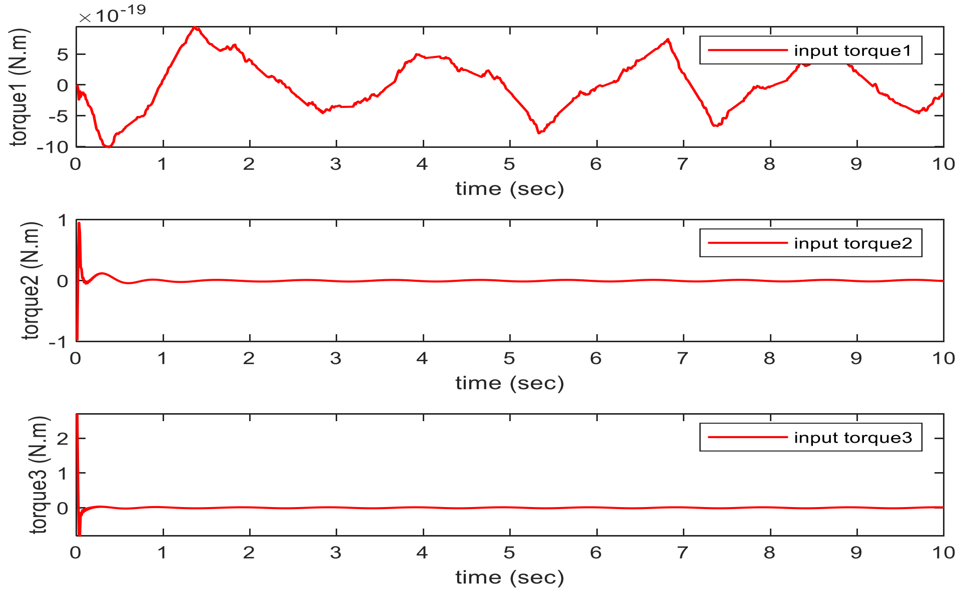

Figure 5-12 Torques applied during simulation

The objective function minimized to obtain optimal parameter values of PD controller is graphed as shown below:

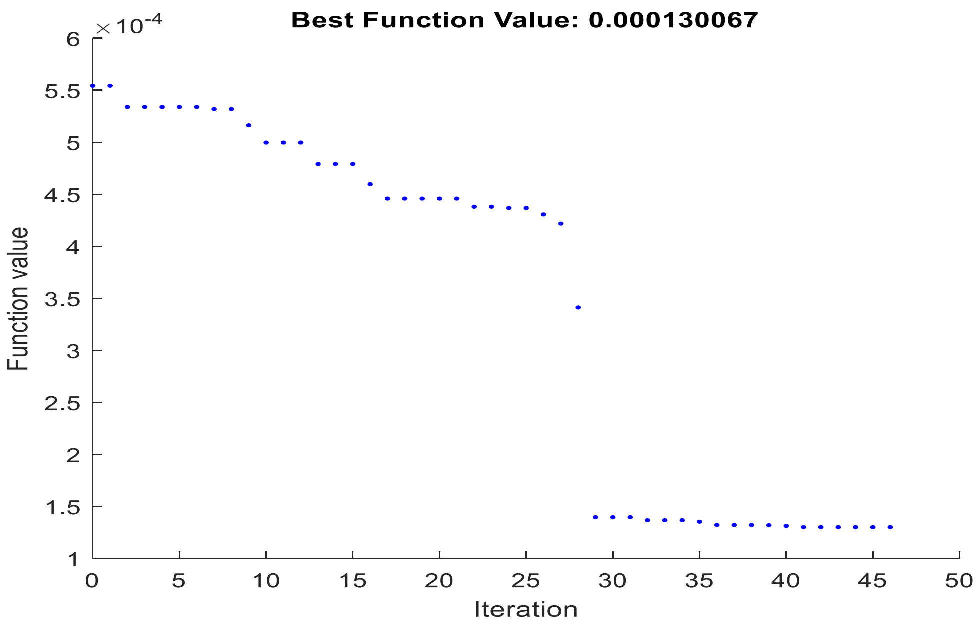

Figure 5-13 Objective function minimized for PD optimal parameters in SLC

Table 5-6 Upper and lower bounds of PSO parameters for PD SLC controller

Parameter	Lower bound	Upper bound
K_p	[1, 1,1]	[1.5,2.5,2.5]
K_d	[0,0,0]	[0.15,0.25,0.15]

Table 5-7 PD parameter values

Gain parameters applied to the leg		
Joints	$\mathbf{K_p}$	$\mathbf{K_d}$
Horizontal hip joint	1.20	0.1
Vertical hip joint	2.4573	0.2
Knee joint	2.350	0.1

CHAPTER SIX

6 CONCLUSION AND FUTURE RECOMMENDATION

6.1 Conclusion

This thesis presented Model predictive control for trajectory tracking of quadruped robot. The controlling action design is categorized in to two part of the system. These are: the trunk balance controller (TBC) and the swing leg controller (SLC) design. The TBC is based on MPC theory and SLC is based on PD controller theory respectively. As per the MPC theory, the TBC fulfills two critical components of the optimized solution and predictive model for following the quadruped's movement trajectory. The TBC was given linear and circular trajectory as a reference to follow and the designed controller have showed best response which is fast convergence, accurate and high steady state performance. Also in the SLC to observe the behaviors of the three angles so that the leg perform stable and proper motion the PD controller was tested by providing reference angles and their respective velocities. The gain parameters of both controllers have been tuned by a metaheuristic algorithm called PSO. Through simulation, this thesis has validated the control method. The simulations for tracking of linear and circular paths of the centroid demonstrate that the handling approach could attain rapid and precise following the intended motion paths, satisfying the control requirements for the quadruped robot. In general, MPC has shown its superiority when compared to PID in improving the steady state performance and tracking accuracy of the quadruped robot.

6.2 Future Recommendations

In this thesis, the designed controllers are used for the system in analytical approach. Numerous areas of opportunity exist for testing the designed controllers on a real robot. Here are some recommendations to make the robot more attractive:

- Design a real quadruped robot and implementing a controller.
- Develop and implement a controller for the legs stance phase.
- Consider mass of payload in the dynamics of the system.
- Use estimation techniques like state observer, kalman filter for estimating the states of the robot dynamic system.
- Control the trajectory tracking for the end-effector of the robot leg.

References

Ammar A. Aldair, A. A.-M. (2019). Design of a Stable an Intelligent Controller for a Quadruped Robot. *Journal of electrical engineering and technology*.

Aranda, R. M. (2022). Control of quadruped leg using analytical approach. *reasearchgate, thesis for obtaining for obtaining the degree of Master's in Optomechatronics*.

B. Katz, J. C. (2019). A platform for pushing the limits of dynamic quadruped control. *2019 International Conference on Robotics and Automation (ICRA),* 6295-6301.

Bledt, G., Powel, M., Katz, B., Carlo, J., Wensing, P., & Kim, S. (1-5 october,2018). Design and control of robust dynamic, quadruped robot. . *In proceedings of the 2018 IEEE/RSJ International conference on intelliegnt robots and systems.* , 2245-2252.

Bledt., G. (2018). Policy regularized model predictive control framework for robust legged locomotion. *PhD thesis, Massachusetts Institute of Technology*.

Boaventura, T., Buchli, J., Semini, C., & Caldwell, D. (2015). Model based hydraulic impedance control for dynamic roobots. *IEEE Trans.Robot.,* 31,1324-1336.

C Semini, N. G. (2011). Design of hyq { a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 225(6):,* 831-849.

Carlo, J., Wensing, P., Katz, B. G., & Kim, S. (1-5 october,2018). Dynamic Locomotion in the MIT cheetah 3 through convex model predictive control. . *In proceedings of the 2018 IEEE /RSJ International conference on intelligent robots and systems* , 7440-7447.

Chen, B., Pei, Z., & Tang, Z. (2016). Adaptive Fuzzy PID control for hydraulic quadruped robot. . *J.Harbin Inst. Technol.,* 48,140-144.

Chen, G., Guo, S., & Hou, B. (2020). virtual model control for quadruped robots. *IEEE Access*, 8, 140736-140751.

Chen, Y., Hou, W., Wang, J., Wang, J., & Ma, H. (28-30 July, 2015). A strategy for push recovery in quadruped robot based on reinforcement learning. . *In proceedings of the 2015 34th chinese control conference* , 3145-3151.

ChengYe Wu, Q. W. (2020). Balance control of quadruped robot based on model predictive control. *Journal of physics*.

Choi, J. (2021). Multi phase joint angle trajectory generation inspired by Dog motion for quadruped for control of quadruped robot sensors. 6366.

D. Wettergreen, H. P. (1995). Behaviour based gait execution for the Dante II walking robot. *In proceedings of the IEEE international conference on intelligent robots and systems(IROS)*.

Dini, N., & Majd, G. (2019). An MPC based two dimensitional push recovery of a quadruped robot in trotting gait using its reduced virtual model. *Mech. Mach.Theory*, 146, 1-25.

Donghyun Kim, J. D. (2019). Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. . *arXiv preprint arXiv:1909.06586*.

Hoang, J., Kim, H., & Kim, S. (2013). Path tracking controller of quadruped robot for obstacle avoidance usong potential functions method. . *Int.J.Sci.Eng.*, 4, 1-5.

Hutter, P. F. (2018). Anymal: a unique quadruped robot conquering harsh environments. *Research Features, (126)*, 54-57.

Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., & Hutter, M. (2019). Learning agile and dynamic motor skills for legged robots. *Sci. Robot*, 4, 831-849.

Kalyoncu, M. A. (2019). Grey Wolf Optimizer Based Tuning of a Hybrid LQR-PID Controller for Foot Trajectory Control of a Quadruped Robot. *Journal of Science*, 674-684.

Kang, R., Meng, F., Wang, L., Chen, X., Yu, Z., Fan, X., . . . Huang, Q. (2021). Bio-Inspired Take-off Maneuver and control in Vertical Jumping for quadruped robot with manipulator. *Micromachines* , 12,1189.

Liu, B., Rong, X., & Chai, H. (2016). A buffering strategy for quadrupedal robots based on virtual model control. *Robot*, 38,659-669.

Luan, F., Na, J., Huang, Y., & Gao, G. (2019). Adaptive neural network control for robotic manipulators with guaranteed finite time convergence. *Neurocomputing* , 337,153-164.

Luo, B. (2022). Balance control based on six dimensional spatial mechanics and velocity adjustment through region intervention and foot landing for quadruped robot. *Robotica*, 89, 1-23.

M. Raibert, K. B. (2008). Big Dog, The rough terrain quadruped robot. *IFAC proceedings volumes, 41*, 10822-10825.

Ma, Z., Liang, Y., & Tian, H. (2020). Research on gait planning algorithm of quadruped robot based on central pattern generator. *In proceedings of the 202039th chinese control conference.*, 3948-3953.

Max Schwenzer, M. A. (2021). Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117:1327–1349.

Ming Lu, B. J. (2022). Design of a small quadruped robot with parallel legs . *Hindawi* .

Muhammed Arif Sen, V. B. (October 2017). Inverse Kinematic Analysis of a Quadruped Robot. *International Journal of Scientific & Technology Research*.

Oliveira, I., Barbosa, R., & Silva, M. (2017). Modeling, Trajectory planning and control of a Quadruped Robot Using MATLAB/SIMULINK. *In proceedings of the ROBOT* , 759-767.

PP Jain, H. P. (2021). Modelling and Simulation of Quadruped Robot using Fusion 360 and Robotic Operating System. *Journal of emerging technologies and innovative research (JETIR),* pg 149-158.

Rodino, S., Matteo Curcio, E., di Bella, A., Persampieri, M., Funaro, M., & Carbone, G. (2020). Design, Simulation and preliminary validation of a four legged robot. *Machines,* 8,82.

Ruchika, N. R. (2013). Model Predictive Control: History and Development. *International Journal of Engineering Trends and Technology (IJETT)* .

S. Hirose, y. K. (1995). Design of a quadruped walking vehicle for dynamic walking and stair climbing. *Advanced Robotics*, 107-124.

Wei Yan, Y. P. (2021). Whole-body kinematic and dynamic modeling for quadruped robot under different gaits and mechanism topologies. *PeerJ Computer Science*.

Xiang Chen, M. S. (2008). Real time software-inthe-loop simulation for control performance validation. *Simulation,* 84(8-9):457{471.

Xianpeng Zhang, L. L. (2015). The quadruped robot locomotion based on force control. *In The 27th Chinese Control and Decision Conference (2015 CCDC),IEEE*, 5440-5445.

Yulong You, Z. Y. (2022). A New Trajectory Tracking Control Method for Fully Electrically Driven Quadruped Robot. *Mdpi machines* , 10, 292.

Yulong You, Z. Y. (2022). A new trajectory tracking control method for fully electrically driven quadruped robot . *Machines* .

Zhang, C., Yang, Z., Liao, L., You, Y., Sui, Y., & Zhu, T. (2022). RPEOD: A real time pose estimation and object detection system for Aerial robot target tracking . *Machines* , 10,181.

Zhang, G., Rong, X., Hui, C., Li, Y., & Li, B. (2016). Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control. *Adv.Robot,* 30, 284-297.

Zhang, G., Rong, X., Li, Y., Cai, H., & Li, B. (2016). Gait control method of quadruped robot diagonal trot based on virtual model robot. 38, 64-74.

Appendix

Appendix A

The developed homogenous transformation matrix is:

$$T_{i-1}^{i} = T_x(\alpha)T_y(\phi)T_z(\theta) = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$$

Where,

$$T_{11} = \cos(\phi)\cos(\theta)$$
$$T_{12} = \sin(\alpha)\cos(\theta)\sin(\phi) - \cos(\alpha)\sin(\theta)$$
$$T_{13} = \sin(\alpha)\sin(\theta) + \cos(\alpha)\cos(\theta)\sin(\phi)$$
$$T_{14} = a\cos(\phi)\cos(\theta) - c\sin(\theta)$$
$$T_{21} = \cos(\phi)\sin(\theta)$$
$$T_{22} = \cos(\alpha)\cos(\theta) + \sin(\alpha)\sin(\phi)\sin(\theta)$$
$$T_{23} = \cos(\alpha)\sin(\phi)\sin(\theta) - \sin(\alpha)\cos(\theta)$$
$$T_{24} = c\cos(\theta) + a\cos(\phi)\sin(\theta)$$
$$T_{31} = -sin\ (\phi)$$
$$T_{32} = \ cos\ (\phi)\ sin\ (\alpha)$$
$$T_{33} = \ cos\ (\alpha)\ cos\ (\phi)$$
$$T_{34} = d - asin\ (\phi)$$
$$T_{41} = \ 0$$
$$T_{42} = \ 0$$
$$T_{43} = \ 0$$
$$T_{44} = \ 1$$

Appendix B

Energy associated with the motion of the three links (Kinematic Energy)

For hip:

$$T_1 = \frac{\dot{\theta}_1^2 (m_1 L_1^2 + 4 I_{xx1})}{8}$$

For thigh:

$$T_2 = \frac{I_{2xx} * \dot{\theta}_1^{\,2}}{2} + \frac{I_{2yy} * \dot{\theta}_2^{\,2}}{2} + \frac{m_2 * (\dot{\theta}_1 * (L_1 * cos(\theta_1) + \frac{L_2 * cos(\theta_1) * cos(\theta_2)}{2}) - \frac{L_2 * \dot{\theta}_2 * sin(\theta_1) * sin(\theta_2)}{2})^2}{2} +$$

$$\frac{m_2 * ((\dot{\theta}_1 * (L_1 * sin(\theta_1) + \frac{L_2 * cos(\theta_2) * sin(\theta_1)}{2}) + \frac{L_2 * \dot{\theta}_2 * cos(\theta_1) * sin(\theta_2)}{2})^2 + \frac{L_2^{\,2} * \dot{\theta}_2^{\,2} * cos(\theta_2)^2}{4})}{2}$$

For calf:

$$T_3 = \frac{I_{3xx} * \dot{\theta}_1^2}{2} + \frac{I_{3yy} * \dot{\theta}_2^{\,2}}{2} + \frac{I_{3yy} * \dot{\theta}_3^{\,2}}{2} + \frac{L_1^{\,2} * m_3 * \dot{\theta}_1^2}{2} + \frac{L_2^{\,2} * m_3 * \dot{\theta}_1^2}{4} + \frac{L_2^{\,2} * m_3 * \dot{\theta}_2^{\,2}}{2} + \frac{L_3^{\,2} * m_3 * \dot{\theta}_1^2}{16} + \frac{L_3^{\,2} * m_3 * \dot{\theta}_2^{\,2}}{8} + \ldots$$

$$+ \frac{L_3^{\,2} * m_3 * \dot{\theta}_3^{\,2}}{8} + I_{3yy} * \dot{\theta}_2 * \dot{\theta}_3 + \frac{L_3^{\,2} * m_3 * \dot{\theta}_2 * \dot{\theta}_3}{4} + \frac{L_2^{\,2} * m_3 * \dot{\theta}_1^{\,2} * cos(2 * \theta_2)}{4} + \frac{L_3^{\,2} * m_3 * \dot{\theta}_1^{\,2} * cos(2 * \theta_2 + 2 * \theta_3)}{16} +$$

$$L_1 * L_2 * m_3 * \dot{\theta}_1^{\,2} * cos(\theta_2) + \ldots$$

$$\frac{L_2 * L_3 * m_3 * \dot{\theta}_1^{\,2} * cos(\theta_3)}{4} + \frac{L_2 * L_3 * m_3 * \dot{\theta}_2^{\,2} * cos(\theta_3)}{2} + \frac{L_2 * L_3 * m_3 * \dot{\theta}_1^{\,2} * cos(2 * \theta_2 + \theta_3)}{4} + \frac{L_2 * L_3 * m_3 * \dot{\theta}_1^{\,2} * cos(\theta_2 + \theta_3)}{2} +$$

$$\frac{L_2 * L_3 * m_3 * \dot{\theta}_2 * \dot{\theta}_3 * cos(\theta_3)}{2}$$

Appendix C

Energy of position of the three links (potential energy)

For hip:

$$U_1 = \frac{L_1 g m_1 \cos(\theta_1)}{2}$$

For thigh:

$$U_2 = -g m_2 (L_1 \cos(\theta_1) + \frac{L_2 \cos(\theta_1) \cos(\theta_2)}{2})$$

For calf:

$$U_3 = -\frac{g m_3 \cos(\theta_1)(2 L_1 + L_3 \cos(\theta_2 + \theta_3) + 2 L_2 \cos(\theta_2))}{2}$$

Appendix D

M matrix of the leg dynamic model:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

Where,

$$m_{11} = I_{1xx} + I_{2xx} + I_{3xx} + (L_1^{\ 2} * m_1)/4 + L_1^{\ 2} * m_2 + L_1^{\ 2} * m_3 + (L_2^{\ 2} * m_2)/8 + (L_2^{\ 2} * m_3)/2 + \ldots$$
$$(L_3^{\ 2} * m_3)/8 + (L_2^{\ 2} * m_2 * cos(2*\theta_2))/8 + (L_2^{\ 2} * m_3 * cos(2*\theta_2))/2 + (L_3^{\ 2} * m_3 * cos(2*\theta_2 + 2*\theta_3))/8 + \ldots$$
$$L_1 * L_3 * m_3 * cos(\theta_2 + \theta_3) + L_1 * L_2 * m_2 * cos(\theta_2) + 2 * L_1 * L_2 * m_3 * cos(\theta_2) + \ldots$$
$$(L_2 * L_3 * m_3 * cos(\theta_3))/2 + (L_2 * L_3 * m_3 * cos(2*\theta_2 + \theta_3))/2$$

$m_{12} = m_{13} = m_{21} = m_{31} = 0$

$$m_{22} = I_{2yy} + I_{3yy} + (L_2^{\ 2} * m_2)/4 + L_2^{\ 2} * m_3 + (L_3^{\ 2} * m_3)/4 + L_2 * L_3 * m_3 * cos(\theta_3)$$

$$m_{23} = (m_3 * L_3^{\ 2})/4 + (L_2 * m_3 * cos(\theta_3) * L_3)/2 + I_{3yy};$$

$$m_{32} = (m_3 * L_3^{\ 2})/4 + (L_2 * m_3 * cos(\theta_3) * L_3)/2 + I_{3yy}$$

$$m_{33} = (m_3 * L_3^{\ 2})/4 + I_{3yy}$$

Appendix E

C matrix in the leg dynamic model

$$C(q,\dot{q}) = \begin{bmatrix} c_{11} & c_{13} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Where,

$$c_{11} = -\dot{\theta}_2 * ((L_2^2 * m_2 * sin(2*\theta_2))/8 + (L_2^2 * m_3 * sin(2*\theta_2))/2 + (L_3^2 * m_3 * sin(2*\theta_2 + 2*\theta_3))/8 + (L_1 * L_3 * m_3 * sin(\theta_2 + \theta_3))/2 + (L_1 * L_2 * m_2 * sin(\theta_2))/2 + L_1 * L_2 * m_3 * sin(\theta_2) + (L_2 * L_3 * m_3 * sin(2*\theta_2 + \theta_3))/2) - (L_3 * m_3 * \dot{\theta}_3 * (L_3 * sin(2*\theta_2 + 2*\theta_3) + 4*L_1 * sin(\theta_2 + \theta_3) + 2*L_2 * sin(\theta_3) + 2*L_2 * sin(2*\theta_2 + \theta_3)))/8$$

$$c_{12} = -\dot{\theta}_1 * ((L_2^2 * m_2 * sin(2*\theta_2))/8 + (L_2^2 * m_3 * sin(2*\theta_2))/2 + (L_3^2 * m_3 * sin(2*\theta_2 + 2*\theta_3))/8 + (L_1 * L_3 * m_3 * sin(\theta_2 + \theta_3))/2 + (L_1 * L_2 * m_2 * sin(\theta_2))/2 + L_1 * L_2 * m_3 * sin(\theta_2) + (L_2 * L_3 * m_3 * sin(2*\theta_2 + \theta_3))/2)$$

$$c_{13} = -(L_3 * m_3 * \dot{\theta}_1 * (L_3 * sin(2*\theta_2 + 2*\theta_3) + 4*L_1 * sin(\theta_2 + \theta_3) + 2*L_2 * sin(\theta_3) + 2*L_2 * sin(2*\theta_2 + \theta_3)))/8$$

$$c_{21} = \dot{\theta}_1 * ((L_2^2 * m_2 * sin(2*\theta_2))/8 + (L_2^2 * m_3 * sin(2*\theta_2))/2 + (L_3^2 * m_3 * sin(2*\theta_2 + 2*\theta_3))/8 + (L_1 * L_3 * m_3 * sin(\theta_2 + \theta_3))/2 + (L_1 * L_2 * m_2 * sin(\theta_2))/2 + L_1 * L_2 * m_3 * sin(\theta_2) + (L_2 * L_3 * m_3 * sin(2*\theta_2 + \theta_3))/2)$$

$$c_{22} = -(L_2 * L_3 * m_3 * \dot{\theta}_3 * sin(\theta_3))/2$$

$$c_{23} = -(L_2 * L_3 * m_3 * sin(\theta_3) * (\dot{\theta}_2 + \dot{\theta}_3))/2$$

$$c_{31} = (L_3 * m_3 * \dot{\theta}_1 * (L_3 * sin(2*\theta_2 + 2*\theta_3) + 4*L_1 * sin(\theta_2 + \theta_3) + 2*L_2 * sin(\theta_3) + 2*L_2 * sin(2*\theta_2 + \theta_3)))/8$$

$$c_{32} = (L_2 * L_3 * m_3 * \dot{\theta}_2 * sin(\theta_3))/2$$

$$c_{33} = 0$$

Appendix F

G matrix in the leg dynamic model

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$$

Where,

$$G_1 = (g * sin(\theta_1) * (L_1 * m_1 + 2 * L_1 * m_2 + 2 * L_1 * m_3 + L_3 * m_3 * cos(\theta_2 + \theta_3) + L_2 * m_2 * cos(\theta_2) + 2 * L_2 * m_3 * cos(\theta_2))) / 2$$

$$G_2 = (g * cos(\theta_1) * (L_2 * m_2 * sin(\theta_2) + 2 * L_2 * m_3 * sin(\theta_2) + L_3 * m_3 * sin(\theta_2 + \theta_3))) / 2$$

$$G_3 = (L_3 * g * m_3 * sin(\theta_2 + \theta_3) * cos(\theta_1)) / 2$$

Appendix G

Developed dynamic model of the quadruped robot:

$$
\frac{d}{dt}\begin{bmatrix}\hat{\phi}\\ \hat{\theta}\\ \hat{\varphi}\\ \hat{x}\\ \hat{y}\\ \hat{z}\\ \hat{w}_x\\ \hat{w}_y\\ \hat{w}_z\\ \hat{\dot{x}}\\ \hat{\dot{y}}\\ \hat{\dot{z}}\\ \hat{g}\end{bmatrix}=\begin{bmatrix}0&0&0&0&0&0&\cos(\varphi)&-\sin(\varphi)&0&0&0&0&0\\ 0&0&0&0&0&0&\sin(\varphi)&\cos(\varphi)&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}\hat{\phi}\\ \hat{\theta}\\ \hat{\varphi}\\ \hat{x}\\ \hat{y}\\ \hat{z}\\ \hat{w}_x\\ \hat{w}_y\\ \hat{w}_z\\ \hat{\dot{x}}\\ \hat{\dot{y}}\\ \hat{\dot{z}}\\ \hat{g}\end{bmatrix}+
$$

$$
\begin{bmatrix}0&0&0&...&0&0&0\\ 0&0&0&...&0&0&0\\ 0&0&0&...&0&0&0\\ 0&0&0&...&0&0&0\\ 0&0&0&...&0&0&0\\ 0&0&0&...&0&0&0\\ -\frac{c_1\sigma_2}{\sigma_1}&-\frac{c_1\sigma_3}{\sigma_1}&-\frac{a_1\sigma_2}{\sigma_1}+-\frac{b_1\sigma_3}{\sigma_1}&...&-\frac{c_n\sigma_2}{\sigma_1}&-\frac{c_n\sigma_3}{\sigma_1}&-\frac{a_n\sigma_2}{\sigma_1}+-\frac{b_n\sigma_3}{\sigma_1}\\ \frac{c_1\sigma_4}{\sigma_1}&\frac{c_1\sigma_2}{\sigma_1}&-\frac{b_1\sigma_2}{\sigma_1}-\frac{a_1\sigma_4}{\sigma_1}&...&\frac{c_n\sigma_4}{\sigma_1}&\frac{c_n\sigma_2}{\sigma_1}&-\frac{b_n\sigma_2}{\sigma_1}-\frac{a_n\sigma_4}{\sigma_1}\\ -\frac{b_1}{I_{zz}}&\frac{a_1}{I_{zz}}&0&...&-\frac{b_n}{I_{zz}}&\frac{a_n}{I_{zz}}&0\\ \frac{1}{m}&0&0&...&\frac{1}{m}&0&0\\ 0&\frac{1}{m}&0&...&0&\frac{1}{m}&0\\ 0&0&\frac{1}{m}&...&0&0&\frac{1}{m}\\ 0&0&0&...&0&0&0\end{bmatrix}\begin{bmatrix}f_{1x}\\ f_{1y}\\ f_{1z}\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ .\\ f_{nx}\\ f_{ny}\\ f_{nz}\end{bmatrix}
$$

Where,

$$\sigma_1 = I_{xx}I_{yy}cos(\varphi)^4 + \ 2I_{xx}I_{yy}cos(\varphi)^2 sin(\varphi)^2 + I_{xx}I_{yy}sin(\varphi)^4$$
$$\sigma_2 = I_{xx}cos(\varphi)sin(\varphi) - I_{yy}cos(\varphi)sin(\varphi)$$
$$\sigma_3 = I_{yy}cos(\varphi)^2 + I_{xx}sin(\varphi)^2$$
$$\sigma_4 = I_{xx}cos(\varphi)^2 + I_{yy}sin(\varphi)^2$$

Appendix H

The product of the skew matrix and the inertia tensor viewed from the global frame:

$$\hat{I}_w^{-1}[r_1]\times = \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ i_{31} & i_{32} & i_{33} \end{bmatrix}$$

Where,

$$i_{11} = \ (c * sin(2 * psi) * (I_{xx} - I_{yy})) / 2$$
$$i_{12} = -c * (I_{xx} - I_{xx} * sin(psi)^2 + I_{yy} * sin(psi)^2)$$
$$i_{13} = b * (I_{xx} - I_{xx} * sin(psi)^2 + I_{yy} * sin(psi)^2) - (a * sin(2 * psi) * (I_{xx} - I_{yy})) / 2$$
$$i_{21} = c * (I_{yy} + I_{xx} * sin(psi)^2 - I_{yy} * sin(psi)^2)$$
$$i_{22} = -(c * sin(2 * psi) * (I_{xx} - I_{yy})) / 2$$
$$i_{23} = \ (b * sin(2 * psi) * (I_{xx} - I_{yy})) / 2 - a * (I_{yy} + I_{xx} * sin(psi)^2 - I_{yy} * sin(psi)^2)$$
$$i_{31} = I_{zz} * b$$
$$i_{32} = I_{zz} * a$$
$$i_{33} = \ 0$$