Synthesis of Continuous Whole-Body Motion in Hexapod Robot for Humanitarian Demining

By

DHAYAA RAISSAN KHUDHER

A Thesis Submitted in Partial Fulfilment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department of Electronic and Computer Engineering
College of Engineering, Design and Physical Sciences
BRUNEL UNIVERSITY LONDON

April 2018

Abstract

In the context of control, the motion of a legged robot is very challenging compared with traditional fixed manipulator. Recently, many researches have been conducted to control the motion of legged robot with different techniques. On the other hand, manipulation tasks have been addressed in many applications. These researches solved either the mobility or the manipulation problems, but integrating both properties in one system is still not available. In this thesis, a control algorithm is presented to control both locomotion and manipulation in a six legged robot. Landmines detection process is considered as a case study of this project to accelerate the mine detection operation by performing both walking and scanning simultaneously. In order to qualify the robot to perform more tasks in addition to the walking task, the joint redundancy of the robot is exploited optimally. The tasks are arranged according to their importance to high level of priority and low level of priority. A new task priority redundancy resolution technique is developed to overcome the effect of the algorithmic singularities and the kinematic singularity. The computational aspects of the solution are also considered in view of a real-time implementation. Due to the dynamic changes in the size of the robot motion space, the algorithm has the ability to make a trade-off between the number of achieved tasks and the imposed constraints. Furthermore, an appropriate hierarchy is imposed in order to ensure an accurate decoupling between the executed tasks. The dynamic effect of the arm on the overall performance of the robot is attenuated by reducing the optimisation variables. The effectiveness of the method is evaluated on a Computer Aided Design (CAD) model and the simulations of the whole operation are conducted using MATLAB and SimMechanics.

Dedication

To the memory of my father Raissan Khudher

To my mother

To my wife and my children Noor, Malak, and Muhammad

To my brothers and sisters

Declaration

I declare that this thesis is my own work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed and reviewed by myself and my supervisors in the Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London UK. All the information derived from other works has been properly referenced and acknowledged.

SIGNED:.. DATE:...

Acknowledgements

Though the following thesis is an individual work, it would not have been possible without the precious support, help, guidance, and effort of many people.

Foremost, I am greatly indebted to my sponsor, the Iraqi Ministry of Higher Education and Scientific Research for offering me the opportunity to undertake this research at Brunel University London.

I would like to express deepest appreciation to my supervisor Dr Maysam Abbod for his comments, advice, endless patience, inspiration, and encouragement, all of which added considerably to my graduate experience.

My sincere thanks also go to Dr. Roger Powell for his expert, sincere and valuable guidance and encouragement extended to me. I would like to thank, Dr Dived Smith, Dr Ayad Khudher, Dr Michael Mistry, Dr Marco Hutter, Zaid Al-Mayyah, Mahdy Mohan, Lulu Yin, Dr Muhammad Usman, Tony Morris and Vinod Sekhri.

Finally but most important, I would like to express my gratitude towards my family for the encouragement, which helped me in completion of this project. My mother, who has been a source of prompting and inspiration to me throughout my life. My beloved and supportive wife, who remains willing to engage in the struggle, and ensuing discomfort, and my lovable children, Noor, Malak, and Muhammad who served as my inspiration to pursue this undertaking.

Contents

List of Figures

List of Tables

List of Acronyms

CAD	Computer Aided Design	ii
CoG	Centre of Gravity	4
CoM	Centre of Mass	26
DH	Denativ-Hartenberg	33
DoF	Degree of Freedom	1
EE	End Effector	6
ID	Inverse Dynamics	69
IK	Inverse Kinematics	36
MTC	Multi-Task Control	11
OSC	Operational Space Control	17
QP	Quadratic Programming	6
TSC	Task Space Control	11

List of Symbols

R_o	Rotation matrix	73
C	*cos*	35
S	*sin*	35
F_i	Ground reaction force	49
G	Gravitational force	38
H	The floating base centripetal, coriolis	38
I_n	Inertia tensor	38
J	Jacobian matrix	35
J_c	Jacobian at contact	38
J_a	Jacobian matrix of actuated joints	66
J_{leg1-6}	Jacobian matrix of of leg 1-6	66
J_u	Jacobian matrix of un-actuated joints	66
M	Inertia matrix	38
S	Selection matrix	39
T	Transformation matrix	35
$T1$	Transformation matrix of leg 1	62
$T2$	Transformation matrix of leg 2	62
$T3$	Transformation matrix of leg 3	62
$T4$	Transformation matrix of leg 4	62
$T5$	Transformation matrix of leg 5	62
$T6$	Transformation matrix of leg 6	62
$T_{CoG,h}$	Transformation matrix between the body and hip	64
$T_{a,fo}$	Transformation matrix between ankle and foot	64

$T_{h,t}$	Transformation matrix between hip and tibia	64
$T_{t,a}$	Transformation matrix between tibia and ankle	64
W	Weighted matrix	69
α	Link twist	33
$\ddot{q}$	Joint acceleration	38
$\dot{q}$	Whole joint velocity	37
$\dot{x}$	Velocity in task space	35, 36
$\dot{\theta}$	Joint velocity	35
μ	Slack variable	71
τ	Actuator torque	38
θ	Joint angle	33
d	Joint distance	33
f_0	Frame at body-hip joint	62
f_1	Frame at hip-tibia joint	62
f_2	Frame at tibia-ankle joint	62
f_3	Frame at ankle-foot joint	62
f_{CoG}	Frame at CoG	62
f_c	Contact forses	38
l	Link length	33
q	Whole joint angles	66
q_a	All actuated joitns	66
q_u	All un-actuated joitns	66
x	Position in task space	33

CHAPTER 1

Introduction

1.1 Introduction

Mobile robots are nowadays dedicated for the automation of many tasks, such as, transport in automated factories, agriculture, demining, and space explorations. The mechanical structure of a mobile robot depends on the nature of the mission to be performed and the the working environment. There are three basic types of the mobile robots: wheeled or track-laying robots, legged robots, and hybrid systems.

Wheeled robots are the most common as they are very effective on even or moderate terrain whereas tracked robots are able to traverse much more varied ground types; but, liked wheeled robots, there is limited ability to choose exactly where to impart their load forces on the ground. Furthermore, both are characteristically 2 Degree of Freedom (DoF) systems (e.g., they can be defined in polar coordinates).

Legged robots are considered to possess superior mobility with an inherent albeit limited ability to change height, thus providing an additional DoF. The contact with the ground is discrete, which allows a selection of the points of support according to the local conditions of the ground. Various applications have been realised, such as inspection or exploration activities in challenging environments. Nevertheless, despite its attractive mobility attributes, using this type of robot is, to date, still very limited in real world. Unlike wheeled and tracked robots, that require a continuous path of support, legged systems have a distinctive characteristic of discontinuous contacts with the ground. In addition, legged robots provide different kinds of locomotion scenarios (gaits), which enable choices of speed/stability to be made not available to other kinds of robots [2]. These attributes enables a legged robot to accomplish tasks in difficult and hazardous environments such as minefields (Figure 1.1).

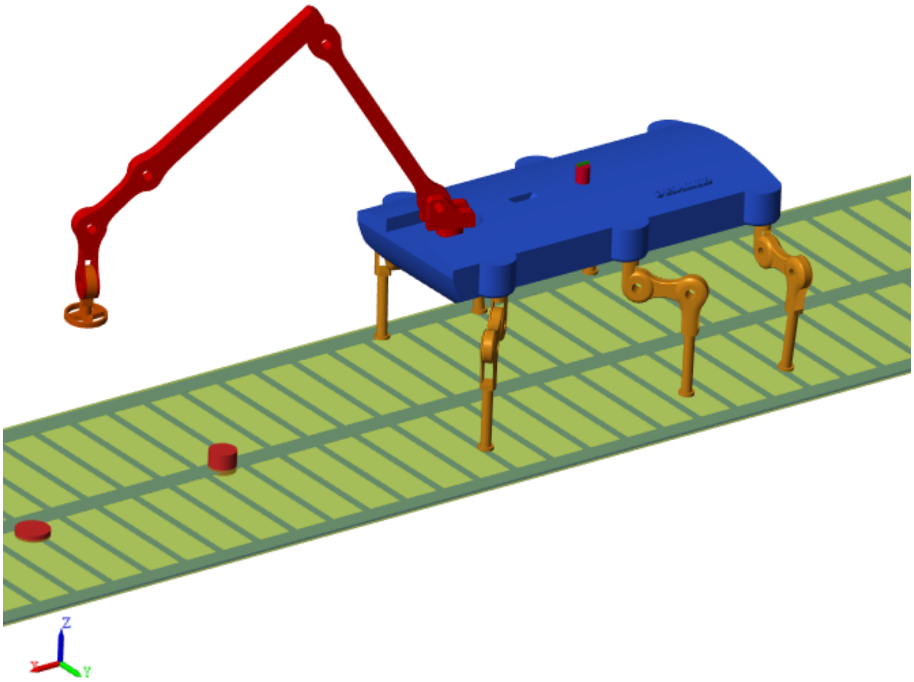


Figure 1.1: Virtual environment of minefield.

The idea of exploiting robots in hazardous areas, such as, minefields has been suggested by many designers [3, 4, 5]. Both legged and wheeled robots are engaged in such duties [6]. Ponticelli et al. [7] has shown that legged robots have advantages over wheeled or tracked robots for navigation through dangerous areas. The ability of legged robot to move with intermittent foot placement allows the robot to manoeuvre over mine or slippage places [8]. The additional DoF means that a legged robot can adapt its body orientation to compensate for terrain gradient [4], and the ability to adjust body attitude while the feet are still on the ground, means that, it becomes an ideal platform for a manipulator arm to be mounted to the body [9]. However, the control of legged robots are very complex; due to the high number of DoF and the dynamical change in the contact state [10, 11].

Obviously, a minefield presents real hazards to human operatives and to robots [12]. Therefore, correct configuration of a robot is very important consideration and must be justified against design criteria. The key components prerequisite in considering hexapod robot for this project, is that this robot can achieve static stability by three legs at each instant of time [13]. The stability of the legged robot is affected by internal and external factors. The internal factors include the kinematics and dynamics of the leg-body system and the selection of gait type [14]. On the other hand, the external factor is the nature of the ground [15]. For the robot to cross over various terrain conditions, it should have proper balance between the manoeuvrability, stability and obstacle overcoming [16]. In general, any leg cannot achieve the locomotion task separately; therefore, cooperation between legs should be considered [17]. According to this reality, two important things will be addressed: the contribution of each leg in the overall motion, and the role of each leg in the body configuration (body balance).

In this thesis, the robot's behaviour (motion and manipulation) is designed according to the tasks that the robot can achieve [18]. These tasks are selected according to the size of the motion space and the importance of the task. The design method relies on dividing the tasks (according to their priority) into different levels [19]. An efficient method that prevent conflict between the tasks will provide an opportunity to perform multiple tasks simultaneously [20]. Indicating that the overall body stability is the highest priority that should be satisfied first [21]. While there are many methods suggested for the stability criteria for legged robots, most of them are based on same fundamental concept, which is to keep the Centre of Gravity (CoG) of the robot inside the supporting legs [22, 23]. For the purpose of using a six-legged robot in minefield, statically stable method would be appropriate as the robot has to walk very slowly [17]. Therefore, accelerating the process depends on performing multiple actions during walking task.

1.2 Motivations

The traditional demining methods are very dangerous and difficult task; any site of previous hostilities is extremely hazardous with many terrains such as bomb craters and many uncertainties as to the whereabouts of ordnance. The problem of landmine would be partially solved if mines could be reliable detected, identified and accurately marked. Although this is a considerable challenge for robotic researches, legged robot might be an effective and efficient means of detecting, and marking mines while guaranteeing the safety of people engaged in the clearing task.

Although there have been many attempts to use six legged robots in dangerous areas; however, these attempts are limited to conventional remote control models. The new trend in control strategies is based on tasks that can be achieved by an autonomous robot. The overall performance of the robot can be improved if all its motion space is used optimally to perform more than one task at the same time.

1.3 Aim and Objectives

The aim of this thesis is to design an autonomous hexapod robot and its control system, with the capability to carry a manipulator arm for minefields scanning, location mapping, and to avoid obstacles. Furthermore, to distinguish the type of terrain and accordingly decide on the gait type and legs configuration. The challenge is how to control the interacting systems of the body movement, body attitude adjustment, manipulator sweep, and leg foothold. Integrating all these characteristics in one legged robot will increase its overall performance. In order to achieve the aforementioned aim, the objectives of this thesis are summarised in the following points:

- Defining the robot tasks using operational space techniques.
- Generating a continuous walking in a six-legged robot by defining the motion reference at the robot's CoG. The reference path includes the desired direction and orientation of the robot's body. The trajectories for each leg is generated and constrained by the robot's body path.
- Tracking a certain straight-line path defined over the ground by the manipulator. The arm can compensate for any changes might happen at the arm-base.
- Reducing the dynamic effect that generated by the manipulator on the robot's body by decomposing the contact forces at each leg to normal forces.
- Evaluating the effectiveness of the algorithm using different motion and scanning scenarios.

1.4 Research Methodology

In order to perform a walking task in a legged robot, the robot's legs must initiate proper contacts with the ground. Part of the available motion space is utilised to satisfy some constraints that arise from the interaction with the environment. The adopted method is to design a CAD model using SolidWorks and modelling all physical parameters, such as, the robot's mass, the moment of inertia and the contact forces. The designed model is imported to MATLAB, and the effectiveness of the algorithm and the robot response are verified using Simulink.

The overall mobility of the robot is achieved by defining set of control points in the task space, such as, the robot's CoG, legs feet, and the manipulator End Effector (EE). The associated constraints are formulated as linear equality constraints (e.g., zero velocity at the contact point) and linear inequality constraints (e.g., joint limit and obstacle avoidance). The reference points are mapped from the task space to the joint space using a Quadratic Programming (QP) solver to ensure handling both types of constraints. In order to prevent the conflict between the executed tasks, a proper decoupling between them is implemented. The approach is based on projecting the tasks with low priority level in the null-space of the higher priority one. The problem of the computational intensity involved with this method is treated by reducing the optimisation variables using QR decomposition method.

Typically, the entire balance is affected when the robot encounters an obstacle inside its path during the walking task; to overcome this problem, the inverse dynamic method is adopted to get a compliant response. During its operation, the manipulator arm produces a dynamic effect on the robot's body, this impact is compensated by controlling the internal force of the robot.

1.5 Contributions to Knowledge

In mobile robotics, continuous locomotion is still generally the domain of wheeled robots, which can operate on paved roads. Although legged robots can walk over uneven terrains, the lack of continuous motion attribute limits their widespread usage. Furthermore, integrating both walking and manipulation at the same time in one legged system is still not common. These two major milestones have been adopted as the essence of this research and the process of detecting landmines using a six legged robot with manipulator arm is considered as a case study in this project. There are many challenges involved in using a robot in hazardous environments, such as, body path planning, legs trajectory generation, planning the arm end-effector path, and handling contact forces between legs and ground. The contributions of this thesis are summarised as the following:

- The overall motion of the robot is achieved by defining a reference path at the robot's CoG and the contribution of each leg to track this path is evaluated. The continuity of the robot's motion is evaluated using different walking scenarios. The velocity of the robot is considered to verify the validity of the approach.

- In order to enhance the scanning operation, a new trajectory planning is designed for the manipulator arm. The trajectory is formed from linear and semi-circular path. This planning provides a significant improvement to the overall robot behaviour in terms of continuity scanning.

- Formulating the problem of mapping the reference path form the operational space to the configuration space as an optimisation problem and solved it using quadratic programming method.

- Integrating both walking and manipulation in one-legged robot by an appropriate exploiting to the joints redundancy of the robot.

- Both the kinematics and dynamics variables are considered in one optimisation problem. The effectiveness of the controller is enhanced by reducing the problem size (eliminating some components from contact forces). The whole body motion is integrated by imposing a proper decoupling between tasks.

1.6 Thesis Outline

This thesis is organised as follows:

- Chapter Two, an overview of legged systems is introduced to focus on the general characteristics of three commonly used types of legged robot. A background to the control strategies used in legged systems is given to explain main aspects in this field. The concepts of task-space, multi-tasks control, and the prioritisation of multi-task control are highlighted in terms of the state of the art. Finally, three aspects, namely, CoG tracking, controlling of force contact, and whole-body behaviour of relevant works are covered.

- Chapter Three, the robot modelling and structure of the adopted robot platform is presented. The kinematics and the dynamics modelling of the robot leg are introduced. The trajectory planning and the influence of trajectory smoothing on the robot performance are addressed. Finally, modelling the contact force between the robot and the ground is detailed.

- Chapter Four, A QP method is exploited to resolve the constrained kinematic redundancy problem. Both inequality and equality constraints are considered explicitly. The problem of redundancy resolution is considered at the inverse differential kinematics level. The CoG of the robot is considered as a control point, and the tracking task is verified by performing three walking modes.

- Chapter Five, a new scanning technique for detecting landmines and unexploded ordnance (UXO) is introduced. In order to speed up the landmine detection and marking, both scanning and moving forward are achieved simultaneously. The robot performs two tasks, the first task is to keep the sensor-head in a fixed level with respect to the ground and the second task is to keep the base of the arm within a specified range of position and orientation. A coupling between the velocity of the end-effector and the velocity of the manipulator base is initiated to ensure the efficiency of the whole process. Four experiments are conducted to verify the validity of the approach.

- Chapter Six, the constraints that arise from the robot dynamic and the interaction with the ground are included in the optimisation problem. The overall performance of the robot is enhanced by decomposing the tangential forces from the contact force components.

- Chapter Seven, conclusions and future works.

1.7 Publications

- Khudher, Dhayaa, and Roger Powell. "Quadratic programming for inverse kinematics control of a hexapod robot with inequality constraints." in Robotics: Current Trends and Future Callanges (RCTFC), IEEE, 2016, pp 1-5.

- Khudher, Dhayaa, Roger Powell, and Maysam Abbod. "Operational space control in hexapod robot for humanitarian demining applications." in Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on. IEEE, 2017, pp 212-216.

- Khudher, Dhayaa, Roger Powell, and Maysam Abbod. "Continuous Landmines Scanning using Legged Robot with Manipulator Arm over Rough Terrain." in Advanced Intelligent Mechatronics (AIM), IEEE, 2017, pp 1495-1500.

CHAPTER 2

Background

2.1 Introduction

Unlike wheeled and tracked systems, where their body balance are controlled passively [23], legged robots have the advantage to control their body balance actively [24]. The intermittent contact with the ground enables the robot to ride or avoid obstacles [25]; these attributes enable legged robots to accomplish tasks in tough and risky environments, such as minefield [7]. However, motion generation for this sort of robots is very challenging; because they own a large number of DoF and their CoG is not actuated [26]. Furthermore, the robot cannot achieve its functionality without satisfying various combinations of constraints [27], such as position limit [28], velocity limit [20], and balance [29].

Transition of the robot's body from one position to another involves performing more than one task. According to how the assigned tasks are accomplished either sequentially or simultaneously, the qualification of a legged robot is determined. There are many actions associated with legged systems: crawling, walking, and trotting [30]; regardless of the travelling style, all these actions require moving many parts at the same time [22].

In this chapter, most aspects related to legged robot are explored. In section 2.2, an overview of the common legged robot is presented, the purpose of this section is to highlight the differences of these robots in terms of structure and design. The general control approaches, used to control legged systems are investigated in section 2.3. The arguments are restricted to three control aspects, namely, *stability*, *gait generation*, and *impact of the environment.* In section 2.4, the concept of *Task Space Control (TSC)* is defined and the requirements for this technique in floating-base system are determined. The concept of *Multi-Task Control (MTC)* is explained and the methods to deal with it are explored in section 2.5. In section 2.6, decoupling techniques are discussed. Some related works are reviewed in section 2.7. Finally, the chapter is summarised in section 2.8.

2.2 Legged Robot Overview

Balancing of any legged robot is a very crucial factor in their functionality [22]. There are three models of stability, namely, dynamic, statically stable[1], and quasi-static stability [31]. For the sake of comparison, three main types of legged robot are considered in this overview, namely, bipedal, quadruped, and hexapod. The purpose of this comparison is to demonstrate the effect of the contact with the ground on the robot control and the overall body balance.

While walking, bipedal robot has less contact with the ground than both four and six legged robots. The motion generation in two legged robot is less affected by the interaction with the environment. The difficulty of the control part lies in the body balance, as the CoG should align with the leg in stance phase [32]. In general, the CoG of the robot follows a sinusoidal shape trajectory. Figure 2.1 shows two legged

[1] The projection of CoG of the robot is always inside supported polygon.

robot tracking a straight line path at its CoG. Any operation that the robot can perform cannot be associated with walking task, as its balance could be affected directly by the performed tasks. The main operations the this robot can perform are valve opening or closing, door manipulating, welding, and drilling; the robot can perform these tasks (that require contact with the environment); but when its legs are in contact with ground to ensure the balance.

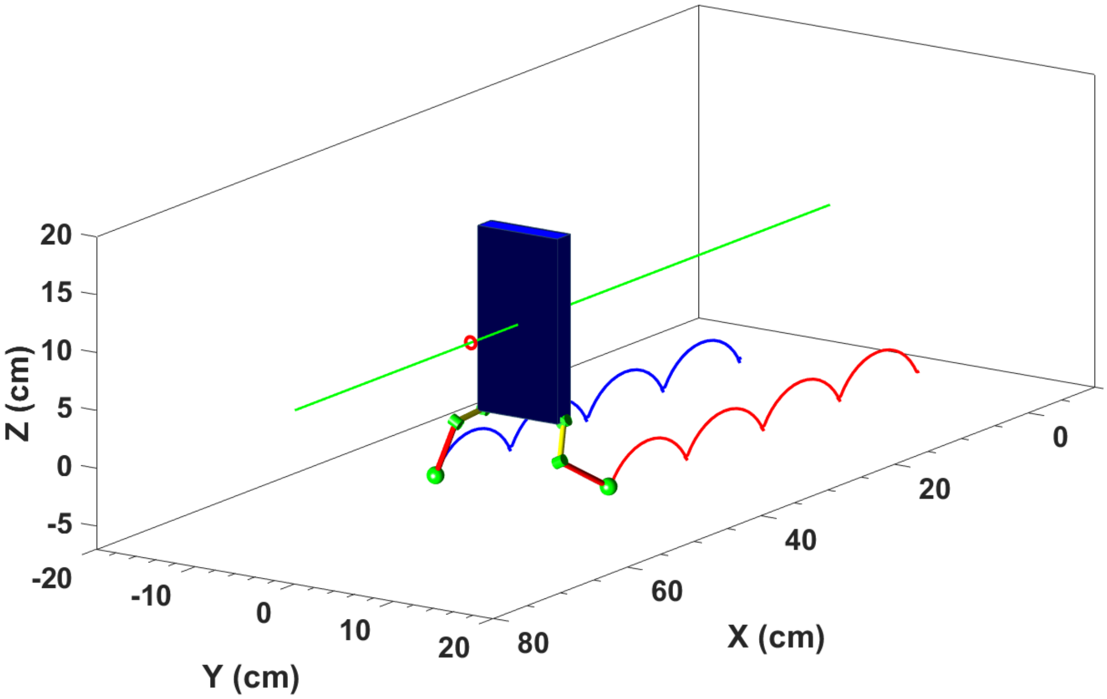


Figure 2.1: Bipedal Robot.

There are many types of two legged robots are dedicated for research and development, such as, ASIMO by Honda [33], ATLAS by Boston Dynamics [34], HRP by AIST [35], and Romeo by Aldebaran [36].

In case of four legged (Figure 2.2), while the robot can perform one cycle by lifting up one leg, the other three legs remain in contact with ground. The static stability walking can be realised by (3/4) duty factor[2]. In terms of distribution of legs under

[2] The proportion of the support phase time to one cycle time.

the body, the design of most quadrupedal robots is inspired from mammals. While this configuration offers a good characteristic from the energy consumption viewpoint [37], the overall body balance is the main concern in controlling this kind of robot especially during walking over rough terrains [23].

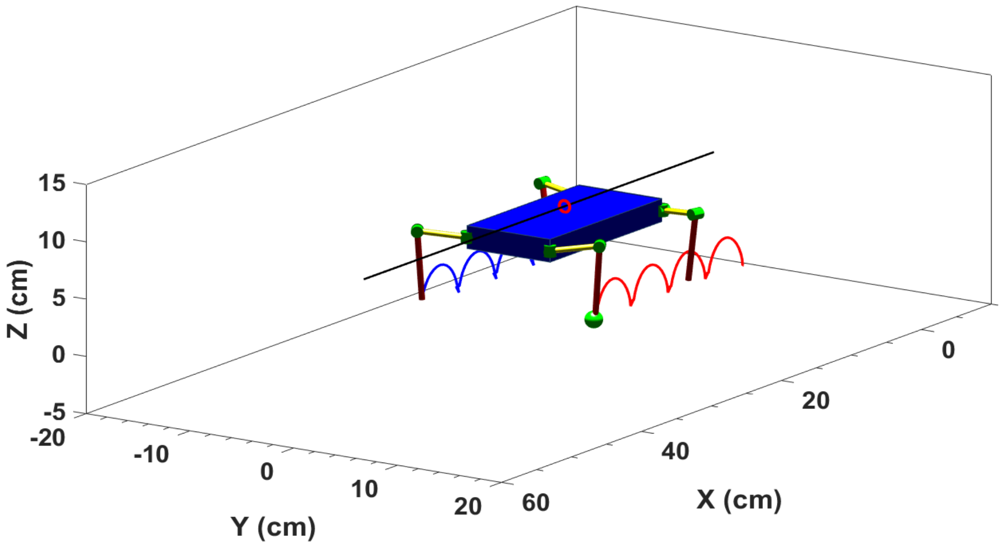


Figure 2.2: Quadrupedal robot.

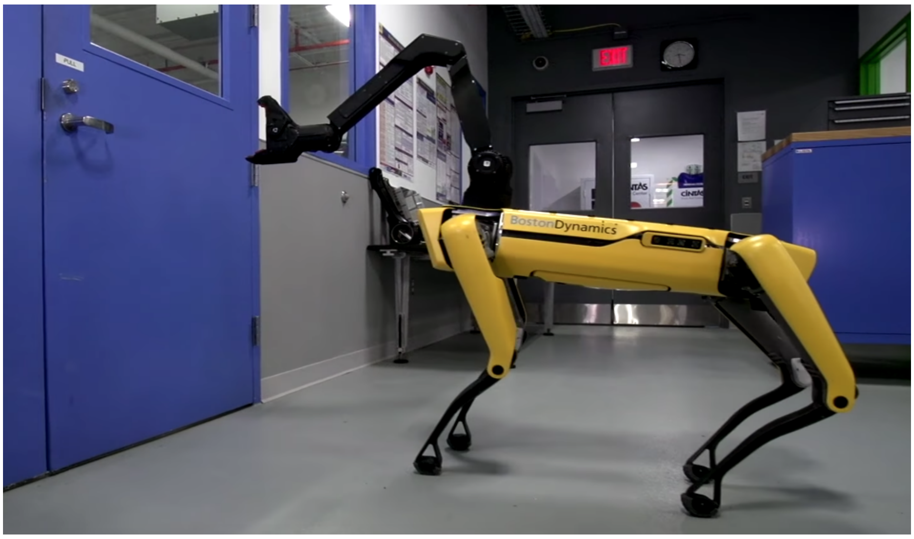


Figure 2.3: SpotMini robot from Boston Dynamics [1].

The latest versions of four-legs robot are LittleDog and SpotMini by Boston Dynamics (Figure 2.3) [1], and StatIETH by ETHzurich [38]. Both of these robots are controlled based on dynamic balancing gait. Although this kind of balance has efficiently been proved in challenging terrain, it is not applicable when negotiating hazardous environments, such as, in minefield.

A six legged robot (Figure 2.4) could potentially negotiate both walking speed and balancing efficiently. The robot can move three legs in each alteration (case of tripod gait) and keep the other three legs on the ground [24]. This characteristic would enhance the robot speed and balance, as the robot can walk with duty factor of 1/2 [14]; the robot can realise different kinds of gait, such as, wave gait [39], and free gait [40]. The stability would be perfectly maintained in case of hexapod robots [41], as the projection of CoG is always inside the supported polygon with any gait type. The aforementioned attributes give the robot a preference over other legged robots, especially in negotiation rough terrains [40]. In the field of hexapod robot, the popular platforms are COMET-IV [42], and SILO-6 [4].

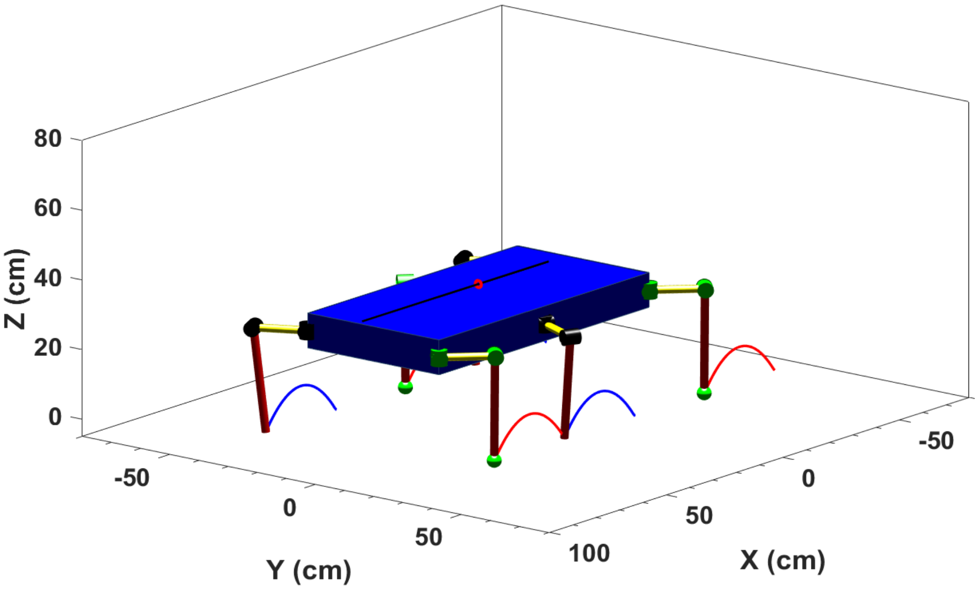


Figure 2.4: Haxapod robot.

2.3 Control of Legged Robots

In the context of legged robot control, there is a vast amount of work addressing the locomotion control [43, 44, 45, 31]. Most earlier approaches are based on the methods that used in fixed-base robots [46]. One paradigm to control the robot position is by considering each leg as an independent manipulator arm, and generating a specific trajectory for each one separately using inverse kinematics methods [24]. The motion of the robot is assured statically by keeping the CoG of the robot inside the polygon created by the legs in support phase [23]. The stability of the body is checked continuously before moving any leg and propelling the robot's body forward, this action is achieved during all legs are on the ground. There are three significant characteristics make the aforementioned method not applicable in six legged robots: no firmly connection to the ground, having more joints than the required to achieve one task, and the motion space is reduced by the interaction with the ground [47]. In the following sub-sections, the attributes that related to this project are discussed in details, such as stability, gait generation, and the impact of the environments.

2.3.1 Stability

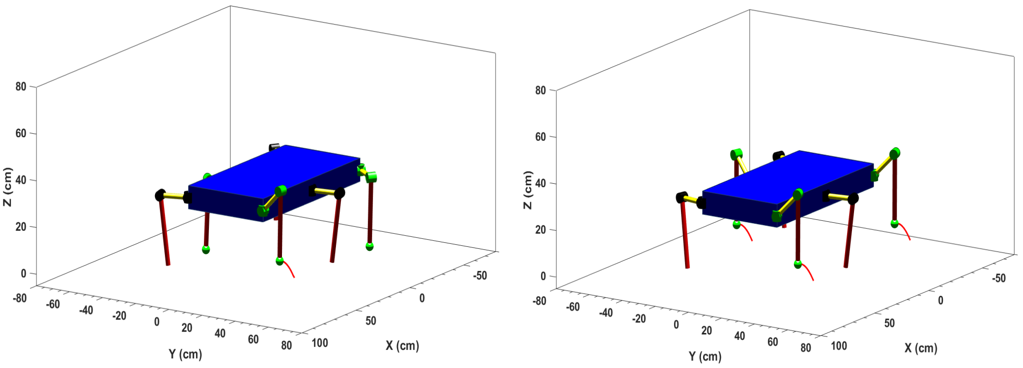
Stability is an essential factor for mobility of legged robots, the robot can fall at any time if the stability is not assured. According to the tasks assigned to the robot and the nature of the environment, the overall stability of legged robots can be achieved statically or dynamically. The necessary condition of statically stable locomotion is retaining the vertical projection of robot CoG inside the polygon that is formed by the legs in contact with the ground. This concept is developed in many researches to increase the probability of stability by defining a new concept called the margin of stability; the idea of this concept is based on the distance between the projection of CoG to the nearest border of the polygon. The essential idea behind this approach is based on the geometric criterion; no dynamic effects, such as, the contact forces and momentum are considered.

On the other hand, the dynamic stability is determined by the supported polygon and the force components generated at the CoG. Depending on the contact type between the robot and the ground, there are two common methods to address dynamic stability: the zero moment point (zmp), which is mainly used in two legs robot, see [48], and considering the robot's CoG as a control point [49, 15, 13]. The contribution of each leg in the overall motion of the robot is calculated according the reference path that prescribed at CoG. Each component of this path contains full information about the robot rotation and translation [50].

2.3.2 Gait Generation

In legged robot, the gait is defined as alternating movements of legs, the robot's body is propelled in a particular direction as a result of this alterations. Typically, in a legged system the gait cycle consists of two phases, namely, stance phase when a leg's foot is in contact with ground and transfer phase when the leg is raised up (no contact with ground). Supporting and driving the robot's body forward are achieved during the stance phase; the foot placement and foothold are accomplished during the transfer phase.

In the field of six-legged robot, there are two main types of gait: the gait with (5/6) duty factor (Figure 2.5a) and tripod gait (1/2) duty factor (Figure 2.5b). The first style of walking provides high degree of stability; typically, this sort of gait is suitable for walking over rough terrains, as there are five legs over ground at the time. The second type permits a trade-off between the stability and walking speed. The sequence of lifting legs depends on the gait design; the sequence of (1, 6, 3, 2, 5, 4) is considered in this project.



(a) Gait with (5/6) duty factor.

(b) Tripod gait.

Figure 2.5: Gait types for a six legs robot.

2.3.3 Impact of the Environment

The problem of physical interaction between a robot and its surrounding has been addressed widely by various approaches. Most these approaches are based on implementing a force feedback in addition to the conventional feedback parameters (position, velocity, and acceleration). The inadequacy of conventional force control method has been specified, in the field of traditional manipulators [51] and legged robots [52].

In order to satisfy the stability and walking continuity, the contact between the robot legs and ground should be compliant; the term *compliant* reflects the ability of a robot to absorb the impact of shocks on the ground. There are two techniques to provide this very important characteristic in legged robots. The first technique is called *passive compliant*, which is based on using springs, dampers, and gearboxes between legs links and joints [25]. There are no actuations involved with this method, and the controller has no role in the robot compliance. On the other hand, the compliant can be actively controlled using *impedance control* technique [53, 54] or *Operational Space Control (OSC)* [55, 56].

2.4 Task Space Control

The key element to develop a versatile and dextrose-legged robot is to design the motion in the space of the tasks instead of the space of the robot's joints [57]. More specifically, this method involves designing the motion in the space of the task to be performed and then mapping the reference to the state-space of the robot's joints [55]. This approach is very appealing, since it offers the ability to accomplish many tasks by the robot simultaneously [58].

The fundamental concept of the task space control is based on partitioning the whole behaviour of the robot into sub-tasks [14]; each sub-task requires some DoF to execute. Hence, an appropriate categorising for all tasks according to their priority and a precise evaluating for the contribution of each joint in the overall operation will enhance the robot performance [59].

In many applications, the reference trajectory is specified as a position and orientation of a certain point in a task coordinate system [60]. Given a task as the reference of motion, the inverse kinematics problem is to find the joint velocity by given the reference velocity in the task-space [46].

Typically, the CoG of the robot can be virtually described as six DoF (three DoF for translation and three DoF for orientation) [27] and the whole path of the robot can be defined at this point. As a consequence of having more DoF, the robot can achieve many tasks in addition to the walking task depending on the robot's state and the constraints [61]. In general, the presence of more constraints (e.g., contact force, velocity limit, position limit) will reduce the dimension of the motion space [62]. Therefore, utilising the robot's redundancy wisely can enhance the ability of the robot to accomplish many objectives [63].

Manipulating more tasks by the robot in addition to the walking task is still limited in ensuring stability [21], decoupling contact force from joint torque [64] or velocity limit [65].

2.5 Multi-Task Control

The concept of multi-task implies the ability of the robot to perform more than one task at the same time [66]. In a fixed base manipulator arm, the robot can achieve one main task by its end-effector, such as, grasping [67], cutting [68], or welding [69]. The robot with more DoF than required to perform a task is called redundant robot; the joints redundancy is necessary when it is required to perform further tasks by the robot [70].

Kinematics redundancy has become increasingly common in robotics in order to enhance the robot achievements [28, 58, 20]. In the field of industrial robots, kinematic redundancy has been used to improve the capability of the robot to avoid obstacles [51], singularity[3] avoidance [71] and joint limit [72].

Legged robots are intrinsically redundant; they have many joints (some are virtually represented by 6 DoF) than the required to perform one task. The robot should handle many sub-tasks in addition to the main task, for instance, tracking CoG path cannot be achieved without maintaining the overall balance of the body and satisfying the dynamic effect of the mechanical parts [73].

The robot's legs play important roles in propelling the body forward, absorbing the impact of the contact [74], and maintaining the body level with respect to ground [13]. On the other hand, the contacts that arise during the walking task have a negative impact on the motion control, as the contacts impose more constraints [75] need to be satisfied by the controller. Depending on their importance, the tasks

[3] Kinematic singularity is a configuration in which a robot loses some DoF. This case happen when the Jacobian matrix becomes rank deficient.

are arranged in different priority level, for example, the body's balance, joint limit, and velocity limit are more important than body posture. Accordingly, the motion space will be used to perform all these tasks in case of consistency between them, otherwise, a proper decoupling between them should be implemented.

Several approaches have been proposed by researchers to solve the kinematic redundancy problem; these are based on defining supplementary tasks to be performed in addition to the main task. The tasks are augmented in one task-space formulation [76, 77]. By assuming that there is no conflict between tasks, this method guarantees that many tasks can be executed at the same time. However, the consistency between tasks cannot be ensured practically without guaranteeing a proper decoupling between tasks. In order to solve the inconsistency between tasks, weighting methods have been proposed by modifying the joints control command according to a certain condition. [78]. The aforementioned methods cannot be used in a complex system, such as, legged robot due to the dynamical changes in the constraints.

The approaches based on projecting the task with low priority level in the null-space of the tasks with high priority level have achieved wide attentions impose a strict decoupling between the executed tasks [79, 80, 81, 82, 83]. Although this method can perfectly decouple tasks in different priority order, it cannot handle tasks that formulated as inequality, such as, joint limit and force bound. Many studies have considered the inequality constraints using the projection methods by converting them it to equality constraints using the potential field approach; this method is based on projecting the gradient of the inequality task to the null space of the main task [84]. Conceptually, this method cannot consider the inequality task in the first level and it suffers from high computational cost.

2.6 Prioritising Multi-Task Control

Prioritisation is a process of arranging many tasks in different priority levels to ensure that the tasks with low priority level are completely decoupled from the tasks with higher priority [85]. In case of some tasks are linearly dependant on one another (case of rank deficient in linear systems) [86], these tasks cannot be accomplished in the same level. Alternatively, in order to consider the utility of the redundancy, the tasks should be arranged in a strict order. In general, the tasks can be formulated as a set of linear equality functions (e.g., zero velocity at the contact) or a set of linear inequality functions (e.g., joint limit), many algorithms are proposed to handle both types of tasks [82]. However, the algorithm that can handle equality and inequality tasks in any level is preferred [87, 88].

In order to take the inequality constraints into account the linear programming (LP) method has been proposed [89, 90, 91]. Although this method offers the ability to handle both type of tasks in any level of priority, finding the optimal solution using LP suffers from two main difficulties. First, due to the complex polyhedron achieved by the simplex algorithm, discontinuities happen when the solutions move from a vertex to other. Second, this method is not beneficial in real time computation [92]. For more blemishes using linear programming see [93].

Instead of the inverse kinematics problem using generalised inverses [94], an alternative approach is to use QP [95]. This method provides considering the inequality tasks in any level of priority [87]. The original QP formulation, both equality and inequality constraints are presented [96].

The QP method has been used widely in robotics for both inverse kinematics [97] and inverse dynamics [98]. While the optimisation variable in inverse kinematics is the robot's velocity [99], the robot's joint torque, joint acceleration, and contact force are the optimisation variables in inverse dynamics [100]. Based on how to consider these three variables directly or indirectly, there are three approaches [101], see Chapter Six for more details.

Recently, promising algorithms have been proposed to handle any type of constraint by weighting or imposing a certain level of priority in the velocity level. Kanoun et al. [87] suggested a method to solve a cascade of QP and find the result in one-step inside the hierarchy. Escand et al. [97] proposed a QR decomposition method to reduce the computation time. In the dynamic level, Saab et al. [62] proposed an algorithm to reduce the computation cost by reducing the force components.

In order to qualify the robot to perform more tasks in addition to the walking task, new constraints should be defined [102]. According to this fact, adding or removing constraints will cause changing the size and structure of Jacobian matrix [57]. Therefore, a trade-off between the number of the achieved tasks and the constraints must be realised [102] and an appropriate hierarchy should be imposed in order to ensure an accurate decoupling between tasks [103]. A proper decoupling prevents the conflicts between the executed tasks [88], and allows full exploitation to the redundant DoF [98].

2.7 Relevant Works

Four main aspects are considered in this work, namely, CoG tracking control, posture control, controlling of force contact, and walking and manipulation. The relevant works are categorised in the following three sub-sections. One more point to take into account here, the structure of the robot is irrelevant, as the task space control is related to a task to be controlled regardless to the type of the robot.

2.7.1 CoG Tracking

There are many attempts to track the CoG of legged robots, starting from the methods that rely on purely static stability methods [4] [13, 49, 12, 104, 105]. In these methods, the body transition is achieved while all legs are on the ground; the robot's balance should be assured before doing any motion. In order to increase the balance condition, an attempt is made to define new stability criterion based on stability margin [5] [13], this method is based on shrinking the original polygon by a certain distance to increase the probability of body balance. By the same token, [49] proposed a method based on swaying the robot body in opposite direction to the leg that will take off. Although, this approach enhanced the stability criterion, the dynamic effects due to the contact forces and the momentum are not considered.

In quadrupedal robot, Buchli et al. [106] reduced the stability criterion to lines created by each diagonal pairs of legs. They proved that the stability polygon is not necessary for quasi-steady state walking. In addition, they developed an efficient way to calculate the CoG path by given foothold position and assuming a zero velocity condition at the contact points. However, this assumption at the foothold is not applicable especially in natural environments unless there is an appropriate decoupling between tasks and joints motion [107].

An interesting work has been done to track the path of the CoG of four legged robot by Kalakrishnan et al. [108]. They generated paths for CoG and the foot in transfer phase using fifth order spline; each segment of a spline is generated according to the body configuration and foothold (no predefined whole CoG path is considered in this approach). The robot speed is enhanced by this method as the phase of body adjustment while all legs are over the ground is eliminated. Although, they used ZMP as a stability criterion, the position of CoG is evaluated every cycle. The

[4] The CoG stay inside the polygon of legs in support.

[5] A shortest distance between the projection of CoG and an edge of supported polygon.

generated CoG trajectory is created instantaneously (according to the CoG and foothold), this implies that the body direction and velocity cannot be controlled directly. There are two assumption associated with this approach: zero velocities at the contact points and the Jacobian of the contact points remains full row rank.

2.7.2 Controlling of Force Contact

The goal of controlling the force contact is to get compliant interactions between the robot and its environment. Exploiting position control techniques based on traditional proportional integral derivative (PID) controller is not enough in the field of legged robot or any application that involves interaction with surroundings. Apart from passive compliant, there are many approaches to gain active compliant, most of them depend on torque as a control command. However, the presence of passive joints in legged systems hinder the direct implementation of standard torque control methods in such robot. Although, there are many attempts to handle this problem using artificial methods [109, 110, 111, 112] or state estimation [113, 114], only model based control will be considered in this review.

There are three main approaches to handle the force between the robot and the contact surface. The first approach relies on eliminating the effect of the contact force; Mistry et al. [64] proposed a method to calculate joint torques by given joints accelerations. In this method the Jacobian matrix is decomposed to orthogonal and upper-triangular matrices using QR decomposition method. Two assumptions were considered in this approach, which are no motion at the contact and the system remains fully constrained. The method was evaluated on bipedal robot to perform squatting (no walking).

Righetti et al. [115] extended the orthogonal decomposition method to include inequality constraints. To ensure that the contact force stays within a certain bound, they defined a linear friction cone around each contact point. The problem of inverse dynamics has been formulated as a QP to minimise both the torque command and contact force.

The second approach is based on operational space control [55]; this method generates a dynamically consistent torque using the inertia matrix as a weighted matrix. In the same sequel, Park et al. [60] has generalised OSC to floating-base robot; the problem size is reduced by difining a 6 DoF virtual joints that describe the relation between the robot and the inertial frame.

The third method relies on considering only the normal force components in the optimisation problem. Saab et al. [62] introduced a solution to handle both equality and inequality constraints in a stack of tasks [6]. The computation cost of this method has been enhanced by formulating the problem without requiring to compute the mass matrix [116]. They calculated the torque that satisfies the contact condition using *Recursive Newton-Euler* algorithm.

2.7.3 Whole-Body Behaviour

Whole-body behaviour is arguably one of the most important characteristics in legged robots to generate agile and dexterous motions. So far, two typical categories have been considered in this field: full-body balance and whole-body motion planning. While the first category involves retrieving the overall balance of the robot during applying an external force, the robot performs walking and manipulation in the operation space in the second category. A short review is conducted in the following to discuss the aforementioned categories.

Sentis et al. [117] constructed the whole-body by integrating various control primitives; three categorise have been defined in this approach: constraints, tasks, and

[6] All tasks are ordered according its priority

posture primitives. A strict hierarchy is established between the tasks to ensure a proper decoupling between them. This framework was implemented on the humanoid robot Asimo [117].

In the same context, Henz et al [118], introduced a new method to combine the robot balance with whole-body control. This method is based on projection of the subordinate tasks in the null-space of the higher priority tasks. The interesting part of this method is that it solves the force distribution part numerically. Furthermore, this approach handles the external perturbation without measuring them. The robot TORO was used to evaluate the validation of this approach [118].

In the field of quadrupedal robot, Winkler et al. [119] have introduced a method based on both inverse dynamic to ensure motions and a virtual mode to account for tracking error by creating a feedback for torque commands. The overall stability has been ensured using ZMP criterion. The method was evaluated on HyQ robot, which is a hydraulically actuated robot.

Kudruss et al. [75] formulated the problem of Centre of Mass (CoM) dynamics in humanoid robots as an optimal control problem. The concept of this approach relies on defining sets of contact models (templates), these sets are used to generate CoM path and dynamically consistent contact force. The technique represents the first obvious application of combining walking and manipulation at the same time. Climbing a stair was the evaluation criteria and this was performed by HRP-2 Humanoid robot. The power consumption of motors has been reduced by 25% using handrail support.

Dai et al. [120] proposed a method to take into account the dynamics of CoM of the robot, which can be represented by angular and linear momentum. The full kinematics model has been considered to satisfy the kinematics constraints, the inverse dynamic has been used to calculate the joint torques.

2.8 Summary

The traditional techniques to control the motion of six-legged robot are inadequate to get satisfying performance for such complicated system. The new trend toward high performance robot is focusing on the tasks as fundamental units in designing the robot's motion. In this chapter, an overview of the main types of legged robot has been presented. Many aspects related to legged robots, such as, the stability, gait generation, and the impacts of the environment have been discussed. The concepts of task control, multi-task control, and the methods to prioritise tasks have been introduced. Finally, a review of literature related to considering multiple tasks in any level of priority has been conducted.

CHAPTER 3

Robot Modelling and Structure

3.1 Introduction

The assigned tasks and the structure of a legged robot play a crucial factor in the design criteria of the robot's motion [22]. The stage of constructing a real legged robot cannot be initiated directly, due to the complexity of the robot; hence, designing a simulation model is very necessary to perform the preliminary tests [121]. Creating a digital prototype for the robot requires defining the robot's kinematics, dynamics, trajectory, and the interaction with environments [4].

Typically, the design should consider all constraints that emerge from the robot kinematics, dynamics, and the operational space [122]. The specifications of the robot, such as, dimensions, shape, legs configuration, the arm structure, and weight have a direct influence on the constraints associated with the robot's motion [123].

The designing phase is started by creating a CAD model of a six legged robot using SolidWorks; then, this model is exported to MATLAB to set all constraints in the corresponding parts. The normal and tangential forces between the robot's legs and the ground impose additional constraints need to be modelled. Simscape multi-body provides several powerful toolboxes to model the contact forces and the forces that generated due to the effect of other legs (the internal forces).

Kinematic modelling includes defining the types of all joints that used in each leg and the length of each link [46]. The method expresses the relation between the adjacent links and the relation between the inertial frame and the frame of end-effector.

The robot's balance and performance are influenced by the quality of the generated trajectories; hence, the concept of the trajectory smoothing is highlighted by considering two types of smoothing techniques: cubic polynomial and quantic polynomial [46].

In this chapter, the robot platform, kinematics, dynamics, the trajectory planning, and the contact force modelling are presented in detail. In addition, some mathematical preliminaries for the proposed approach are addressed.

3.2 Six legged Robot Platform

Constructing a real six-legged robot is very expensive especially when using torque controlled actuators (the robot needs 24 actuators in total), sensors (e.g., encoders,force sensors, and gyroscopes), and controllers (PID controller). Therefore, all the preliminary evaluations are achieved using a CAD model designed in SolidWorks. The designed robot is dedicated to demining applications, therefore, three design requirements have been considered in this model, namely, shape, weight, and size of the robot. The rectangular shape has been chosen for two reasons: it gives the ability to walk along a straight-line easily and the gait of each leg could be the same and to provide enough space for the manipulator arm to scan the area in front the body. The main reason behind choosing a certain weight and shape for the robot is to minimise the probability of detonating a mine in case of unintentional stepping over it. Additionally, the specified size enables the robot to navigate in small spaces.

Figure 3.1 shows a CAD model of a six-legged robot; the figure gives an approximate visualisation of the robot dimensions with respect to human. Each leg has three joints, which are ordered similar to reptile's legs. The rotation axis of the first joint is about the normal coordinates; the rotation axis of two other joints is shifted by 90° around the vertical direction. The leg is consists of three links, namely, hip, tibia, and ankle. The hip joint represents the connection part between the robot's body and the leg. The length of the links is 0 cm, 15 cm, and 30 cm for hip, tibia, and ankle link respectively. The length of ankle link is chosen to provide an appropriate distance to isolate the body of the robot from ground.

The robot is equipped with manipulator arm to carry a sensor head at its end-effector for detecting land mines and any object that may be in the way of the robot. More details about the specifications of the arm will be presented in Chapter 5.

In general, the robot looks like a trunk with seven limbs. The full body has 24 DoF (6 × 3 DoF) for the legs and (1 × 6 DoF) for the arm. In order to consider the physical parameters of the robot, the CAD software automatically calculates physical parameters such as mass and inertia tensor. In addition, it provides powerful tools to visualize the developed model. Table 3.1 illustrates the physical specifications of the robot body and legs. To perform various virtual experiments and to obtain required outputs, the CAD model was imported to MATLAB / SimMechanics environment. Simscape Multibody has a powerful environment to simulate multi-body systems. It has different types of joints, actuators, framing system, and sensors. Many types of controller can be implemented in this environment such as position control and torque control. Additionally, it has its own simulator, therefore, no need to use an external simulator for instance (gazebo or V-REP).

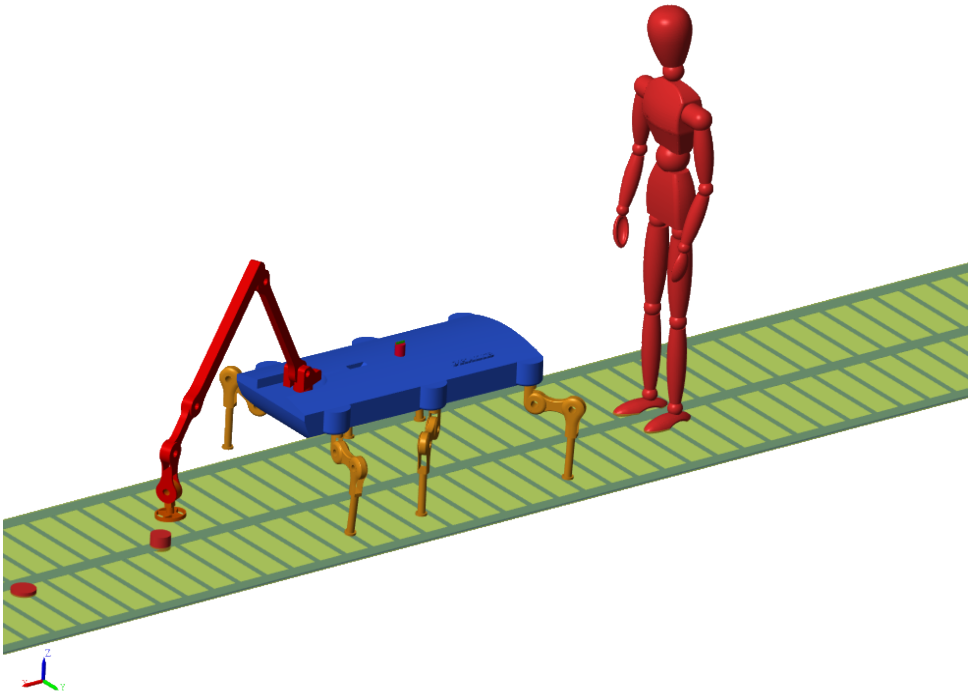


Figure 3.1: CAD model of hexapod robot using SoildWorks. The figure shows an approximate visualization to the dimensions compared to human size.

Table 3.1: The specification of the robot.

Part	Dimension (cm)	Weight (kg)
Body	111 (length) × 85 (width) × 50 (height)	20
Hip	15	1
Tibia	20	1
Ankle	25	1

The robot is controlled using two controller: inverse dynamic controller to provide a feed-forward torque control command and a PID controller to compensate for any error between the reference input signal and the measured signal. While the first controller provides a low gain feedback control command, the second one supply a high feedback gain; a compliant response is achieved by combining these two controller. The basic idea of this type of controller is by applying the reference trajectory that consist of the variables (joint position q_{des}, joint velocity $q_{\dot{des}}$, and joint acceleration $q_{\ddot{des}}$); the measured joint position q and velocity $\dot{q}$ of the robot are supplied back to both controller to generate the torque command. The Simulink environment for the PID controller and the inverse dynamic controller are presented in Appendix C.

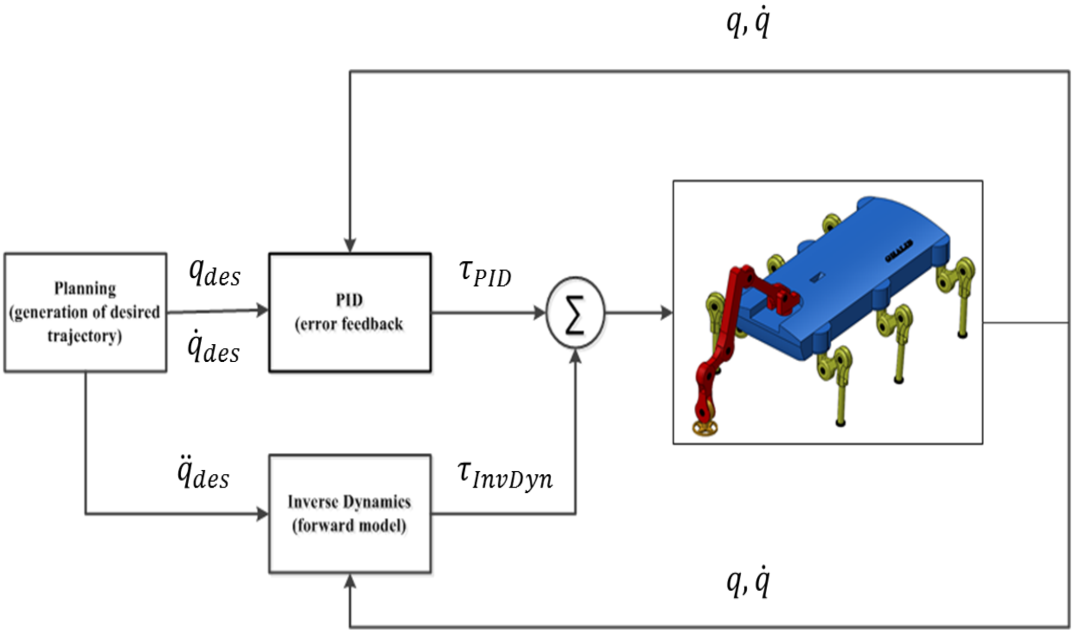


Figure 3.2: Block diagram of the robot control structure. The output of the inverse dynamic and PID controller are denoted as τ_{invDyn} and τ_{PID} respectively.

3.3 Forward and Inverse Kinematics

The forward kinematics define the position and orientation of the end-effector of an articulated body with respect to a base coordinate frame [46]. The position and the orientation of the end-effector can be represented as a function of the joint angles of each individual link; this relation can be mathematically expressed as:

$$\boldsymbol{x} = f(\boldsymbol{\theta}) \tag{3.1}$$

where x is the position and orientation of the foot-tip and θ represents the angles of the leg joints.

The standard method to representing the kinematics model of any link series is the Denativ-Hartenberg (DH) method. This method is based on assigning a frame to each joint, and then defining the relation between adjacent frames [124]. Each frame is identified by two joint parameters (joint rotation and joint translation) and two link parameters (link's length and link's twist) [125]. The joint variables θ are needed to describe the joint rotation (joint angle in the case of revolute joint), or represent the joint translation d (joint distance in the case of prismatic joint). The link parameters are the link length l and the link twist α, which represents the rotation of a certain link around x-axis. While there are no prismatic joints in the considered leg, the value of joint distance is fixed to zero and the values of $\boldsymbol{l}$ remain constant. For a leg with three DoF, there are four frames, as shown in Figure 3.3. The frame (x_0, y_0, z_0) represents the base frame, which is attached to the robot's body. The last frame (x_3, y_3, z_3) represents the robot's foot. Table 3.2 lists the DH parameters of one leg of the robot.

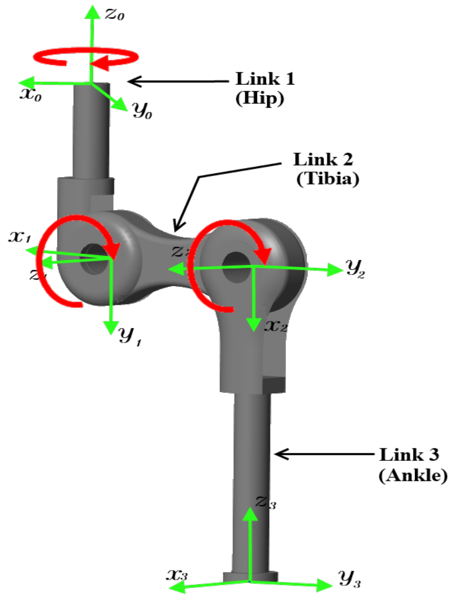


Figure 3.3: Coordinates definition for defining DH parameters presented in Table 3.2.

Table 3.2: D-H parameters of three joints leg.

One Leg				
Link	l_i	α_i	d_i	θ_i
1	l_1	-90	0	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	θ_3

The relative translation and rotation between *ith* and *i*-1 coordinate systems (adjacent links) can be represented as a homogeneous transformation matrix:

$$T_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & l_i C\theta_i \\ S\theta_i & C\theta_1 S\alpha_i & -C\theta_i S\alpha_i & l_i S\theta_i \\ 0 & S\alpha_i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

where T is the transformation matrix, C and S are short for *cos* and *sin* respectively.

The leg-foot reference frame 3 can be expressed with respect to the leg-base frame 0 as:

$$T_3^0 = T_1^0 T_2^1 T_3^2 = \prod_{i=1}^{3} T_i^{i-1} \tag{3.3}$$

$$T_3^0 = \begin{bmatrix} C\theta_1 C(\theta_2+\theta_3) & -C\theta_1 S(\theta_2+\theta_3) & S\theta_1 & p_x \\ S\theta_1 C(\theta_2+\theta_3) & -C\theta_1 S(\theta_2+\theta_3) & C\theta_1 & p_y \\ S(\theta_2+\theta_3) & C(\theta_2+\theta_3) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

The position of the leg-foot (p_x, p_y, p_z) can be represented with respect to leg base frame as:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} (l_1 + l_2 C\theta_2 + l_3 C(\theta_2+\theta_3))C\theta_1 \\ (l_1 + l_2 C\theta_2 + l_3 C(\theta_2+\theta_3))S\theta_1 \\ l_2 S\theta_2 + l3S(\theta_2+\theta_3) \end{bmatrix} \tag{3.5}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{3.6}$$

where $\dot{x}$ is the velocity at task-space, J is the leg Jacobian matrix that map the position and the orientation of the foot-tip to the joint-space, and $\dot{\theta}$ is joints velocity.

Due to non-linearity, including rotation at the joint translation, the problem can be made linear by differentiating Equation (3.1) with respect to θ, obtaining the velocity differential model as in Equation (3.6). This equation describes the relationship between the velocity (angular and linear) of EE and the joint velocities, which is essential in velocity level control, as shown in Equation (3.6).

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial P_{\theta_1}} & \frac{\partial P_x}{\partial P_{\theta_2}} & \frac{\partial P_x}{\partial P_{\theta_3}} \\ \frac{\partial P_y}{\partial P_{\theta_1}} & \frac{\partial P_y}{\partial P_{\theta_2}} & \frac{\partial P_y}{\partial P_{\theta_3}} \\ \frac{\partial P_z}{\partial P_{\theta_1}} & \frac{\partial P_z}{\partial P_{\theta_2}} & \frac{\partial P_z}{\partial P_{\theta_3}} \end{bmatrix} \tag{3.7}$$

$$J = \begin{bmatrix} (l_1 + l_2 C\theta_2 + l_3 C(\theta 2 + \theta 3))S\theta 1 & -(l_2 S\theta_2 + l_3 S(\theta 2 + \theta_3))C\theta_1 & -l_3 S(\theta_2 + \theta_3)C\theta_1 \\ (l_1 + l_2 C\theta_2 + l_3 C(\theta_2 + \theta_3))C\theta_1 & -(l_2 S\theta_2 + l_3 S(\theta_2 + \theta_3))S\theta_1 & -l_3 S(\theta_2 + \theta_3)S\theta_1 \\ 0 & l_2 C\theta_2 + l_3 C(\theta_2 + \theta_3) & l_3 C(\theta_2 + \theta_3) \end{bmatrix} \tag{3.8}$$

$$\boldsymbol{\theta} = f^{-1}(\boldsymbol{x}) \tag{3.9}$$

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{x}} \tag{3.10}$$

where $\dot{\theta}$ is the joint velocity and $\dot{x}$ is the foot-tip velocity.

The elements of the Jacobian matrix are described in Equations (3.7) and (3.8). The problem of computing the joint variables for a given position and orientation of the EE is termed Inverse Kinematics (IK). This situation is fundamental in OSC. In other words, derivation of IK equation is essential for foot placement, trajectory planning, obstacle avoidance, and singularity robustness. In order to calculate θ with respect to x the inverse of Equation (3.1) is required as shown in Equation (3.9).

In order to solve the linear system of Equations (3.6) when $\dot{x}$ is known, it is necessary to invert the Jacobian matrix J, as shown in Equation (3.10).

However, in many situations, there are several difficulties associated with solving the IK problem, particularly when considering the operational space constraints, such as obstacles. The problem of inverse Jacobian matrix can be exacerbated when the space of joint is greater than the space of task (case of redundancy), because there are more than one solution. Another situation when a robot loses some DoF, the Jacobian matrix will become rank deficient (case of singularity).

Several researchers have addressed the problem of IK and calculation of Jacobian matrix using weighted pseudo-inverse [86] and damped least-squares methods [126]. These methods entail several difficulties when applied in highly redundant robots such as six-legged robot. These approaches give no guarantee to implement different types of constraint [127]. They are time consuming in terms of computation [96]. Using traditional methods will become prohibitive due to high complexity of the mathematical structure of the formulation. Therefore, IK problem can be solved using numerical algorithms after formulating it as an optimisation problem [128].

In addition to the above difficulties, exploiting IK in controlling legged-system is hindered by the presence of passive joints, which describe the position and the orientation of the robot body [129]. Furthermore, as legged robots have abundance in the number of joints; it is convenient to take advantage of redundancy in achieving multiple tasks. However, a proper decoupling between executed tasks is compulsory. One of the interesting methods of decoupling tasks is by projecting the secondary tasks in the null-space of the more important tasks. Therefore, Equation (3.10) can be generalised to include the null-space of Jacobian matrix as indicated below for two tasks.

$$\dot{q} = J\dot{x}_1 + (I - J^{-1}J)\dot{x}_2 \tag{3.11}$$

where $\dot{q}$ is the velocity in the joint-space, $(I - J^{-1}J)$ is a projector in the null-space of J, $\dot{x}_1 \in \Re^m$, $\dot{x}_2 \in \Re^{n-m}$.

As this method provides a proper decoupling between tasks and could be utilised as long as there is enough space. However, this is not in case of implementing both unilateral and bilateral constraints. Therefore, in this thesis, the IK problem is derived as an optimisation problem and solved using QP as explained later in Section 4.4 and Section 6.4.

3.4 The Robot Dynamics

The values of dynamic parameters of the robot, such as CoM position, inertia (I_n), and contact forces are essential to formulate the control problem [130]. Calculating these parameters from the mechanical structure is not straightforward. Therefore, modelling the robot using CAD system will allow getting the values of the dynamic parameters. A complete dynamics model of a realistic hexapod robot is required to analyse and compensate for the dynamic effect. In this project, Lagrange formulation is used to model the dynamic parameter [131]. Full derivation of the robot equation of motion is shown in Appendix A. The dynamic model of the robot without contact force is described as:

$$M(q)\ddot{q} + H(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.12}$$

where $M \in \Re^{(6+n)\times(6+n)}$, inertia matrix of the system, $\ddot{q} \in \Re^{n\times 1}$is the joint acceleration, $H\ (q,\dot{q}) \in \Re^{(6+n)\times 1}$ is a vector of Coriolis and centrifugal forces, $G\in \Re^{(6+n)\times 1}$ is vector of gravity forces, $\tau \in \Re^{(n\times 1)}$ is vector of joint torques. Equation (3.12) represents the system in free space (no contact force) and without passive joints (the system is totally actuated). However, legged robots consist of two crucial factors that should be implemented in its model.

$$M(q)\ddot{q} + H(q,\dot{q})\dot{q} + G(q) + {J_c}^T f_c = S^T\tau \tag{3.13}$$

where J_c is the Jacobian at contact point, f_c is the contact force (normal force and friction force), $S = [I_{n\times n}\, 0_{n\times 6}]$ is the selection matrix, the first six elements of this matrix are selected equal to zero to allow to represent the passive 6DoF joint of the robot body.

In case of floating base robot, such as, a six legged robot, the contact forces components between the robot's legs and ground and the passive joints at CoM of the robot should be included in the equation of motion, as shown in Equation (3.13).

3.5 Trajectory Planning

There are two main considerations make planning the trajectory for a legged robot is very complex [22]. First, the base of legged robot is inherently not fixed to ground and consequently, the operation workspace cannot be specified, as it depends on the robot instantaneous position and orientation. Second, depending on its structure, legged robots have more than one limb (six legs and one manipulator in this case), a synchronisation should be implemented between all limbs.

The essential problem of trajectory planning is to allocate a motion modularity for the robot to move from a particular starting place to some required ending place (in some cases, it is required to acquire a specific final configuration). In general, trajectory planning requires three important issues [46]. First, defining detailed information about the starting point, ending point, and the intermediate points; these points can be predefined, in case of an obstacles-free path. Second, satisfying the constraints that arise from the configuration space (joints limit, velocity limits, and joints torque) and from the operational space (avoiding obstacles and following specified path geometry). Finally, the dynamic effects of the robot physical parts (joint stiffness and joint damping).

The simplest type of path planning is between two points sometime called point-to-point path [132]. Typically, in addition to zero velocity at both ending, the requirements of this type of path planning are the position of starting and ending point. Consequently, there is no concern about the way that the EE reached the final point. However, this approach is not applicable in the case of multi-limb robots or when the operational space imposes constraints should be satisfied [133]. Therefore, in this thesis, the trajectory planning will consider creating intermediate points between the initial and final point. The intermediate points are defined according to the tasks that the robot performs and the robot configuration. The sequence of these points will constitute the whole path considered as a reference input.

Three main trajectories are considered, namely, CoG, feet, and the trajectory of the manipulator EE. Different constraints are imposed depending on the trajectory of corresponding limb, for instance, the trajectory of robot's feet consist of force constraints must be satisfied. The trajectory of the manipulator has constraints of avoiding obstacles inside its path. The overall position and orientation of the robot are confined the trajectory of the robot's CoG. Consequently, generating the trajectory of both EE and feet-tips are depending on the CoG trajectory. The results of trajectory generation are the position, velocity, and acceleration of any frame with temporal qualities [46]. These three parameters are described by a time sequence to define the position and orientation (pose) of a controlled point in space.

In terms of implementing many kinds of constraints related to the task space, the operation space method is very useful to generate trajectory [134]. This method is based on exploiting the inverse kinematics techniques, by given a pose of any points in space the corresponding values in configuration space can easily obtained [49]. However, this method is hard to apply in case of singularity and exist of joint redundancy [135]. Having achieved the configuration of the robot related to each path point, the trajectory in the configuration-space is interpolated using an interpolation method to get smooth trajectory [34]. The modularity of the generated trajectory has a direct effect of the overall performance of the robot [136], as will be shown later on.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{3.14}$$

where q represents the position in the joint space, and the value of $a_j, j = 1, 2, 3$ are a constant numbers determined using a set of boundary conditions and t represents the time.

According to the desired continuity level of the position, velocity, and the acceleration, there are two main ways to interpolate a sequence of predefined points [125], namely, cubic polynomial, and quantic polynomial. When the desired trajectory is with velocity parabola and linear acceleration, the cubic polynomial method will be enough, as in Equation (3.14). Four conditions should be implemented before applying this method, which are the positions and velocities at both end of a path primitive[1]. For more details about setting the constraints and solving for the coefficients ($a_0, a_1, a_2,$ and a_3), see [124].

Figure 3.4 shows the trajectories of the robot's hip joint, tibia joint, and ankle joint that created by cubic polynomial.

[1] Segment of path with certain configuration.

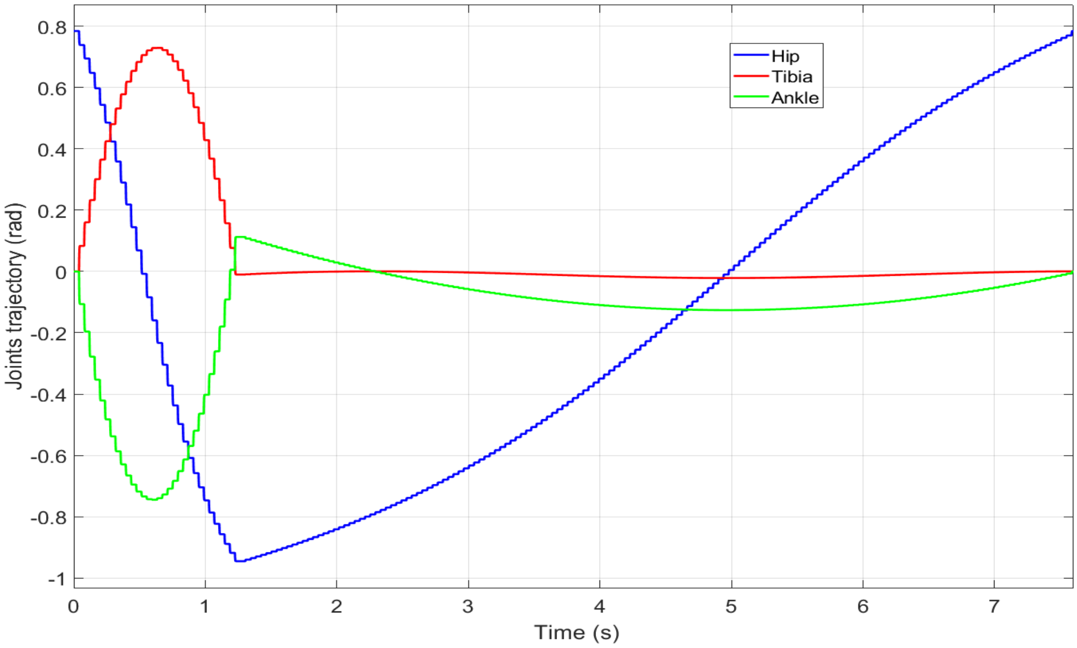


Figure 3.4: The trajectories of the robot's hip joint, tibia joint, and ankle joint. The trajectory is generated by a cubic polynomial.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{3.15}$$

where q represent the position in the joint space, and the value of $a_j, j = 1, 2, 3, 4, 5$ are a constant numbers determined using a set of boundary conditions and t represent time.

On the other hand, when the continuity in the acceleration level is required in planning a trajectory, higher-order polynomial should be considered [125], such as quantic polynomial, Equations 3.15. This is a prerequisite when the robot has to avoid resonances due to contact with the environment or the dynamic effect such as friction [137] In this method six conditions have to be satisfied, as shown below.

$$
\begin{aligned}
q_0 &= a_0 \\
q_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\
\dot{q}_0 &= a_1 \\
\dot{q}_f &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\
\ddot{q}_0 &= 2a_2 \\
\ddot{q}_f &= 2a_2 + 6a_3 t_f + 12a_{4t_f^2} + 20a_f t_f^3
\end{aligned}
\tag{3.16}
$$

where q_0 is the initial position, q_f is the final position, $\dot{q}_0$ is the initial velocity, $\dot{q}_f$ is the final velocity, $\ddot{q}_0$ is the initial acceleration, and $\ddot{q}_f$ is the final acceleration.

In the case of a sequence of multiple of intermediate points each point with contineous velocity and acceleration more constraints should be implemented [138]. Figure 3.5 shows the trajectories of hip joint, tibia joints, and ankle joint of leg 1. The trajectories have been generated using quintic polynomial.

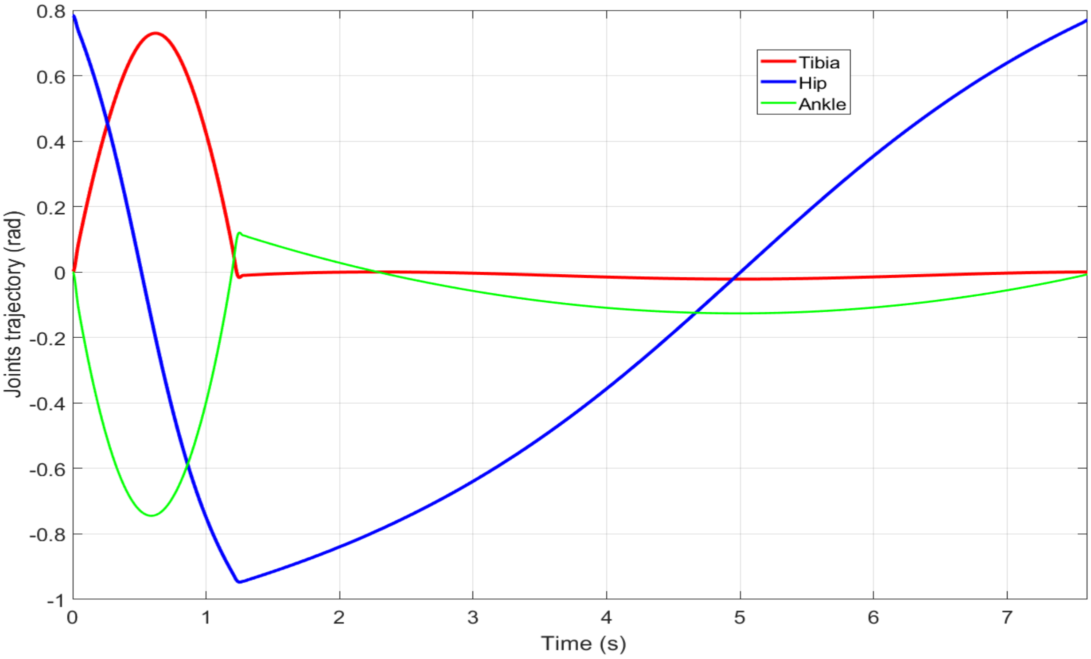
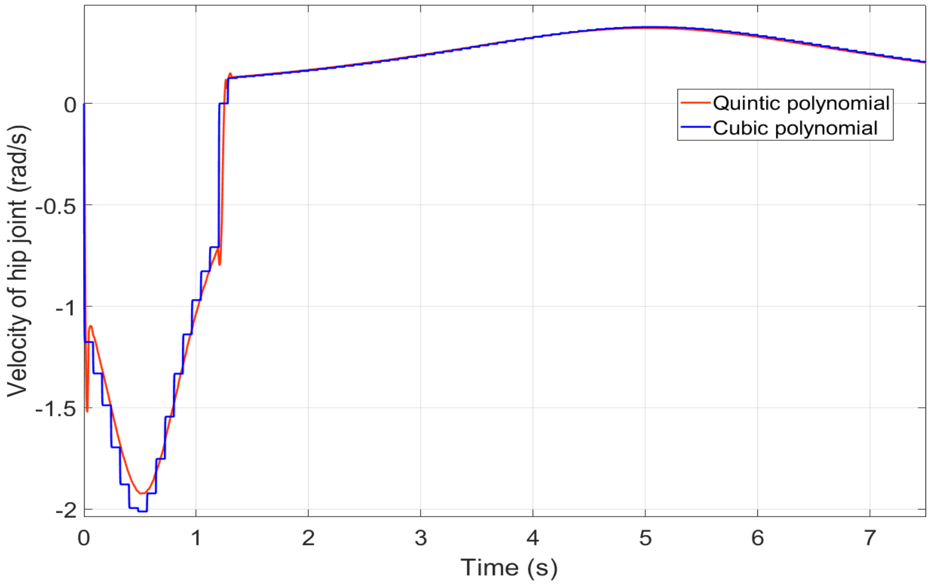
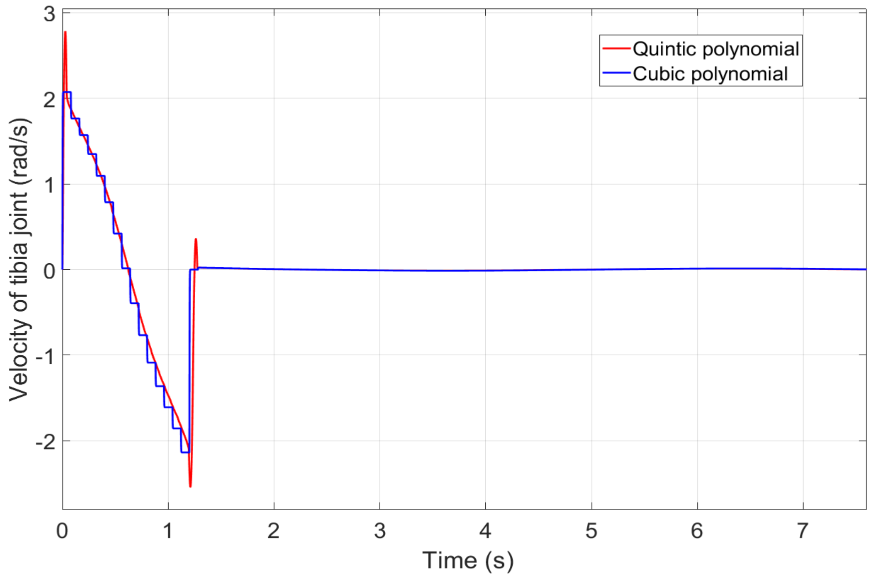


Figure 3.5: The trajectory of the robot's hip joint, tibia joint, and ankle joint. The trajectory is generated by a quintic polynomial.

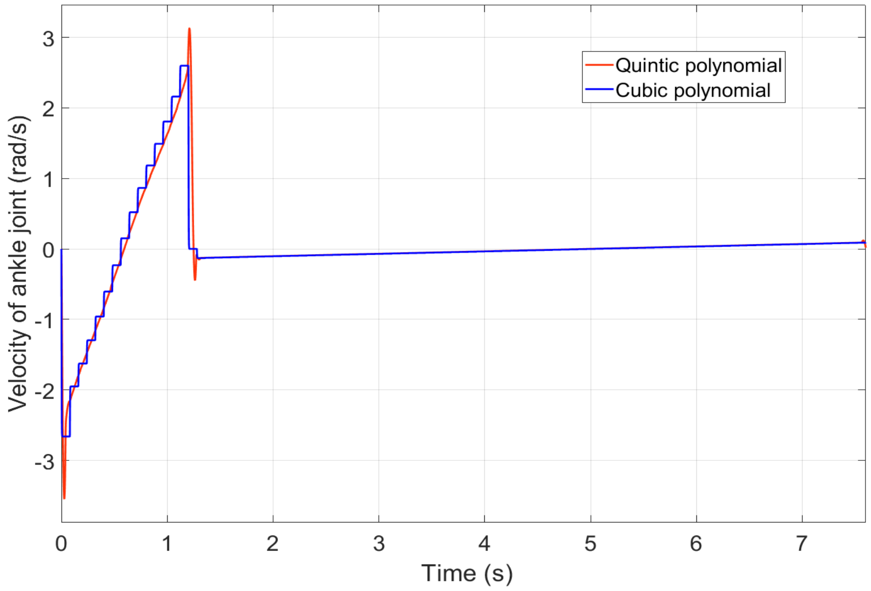
As anticipated, the trajectory smoothing is further affecting to the generated velocities at each joint. Figure 3.6 shows the velocities of hip, tibia, and ankle joints of leg 1. It is clear from the figure that the generated velocities of leg 1 joints using quantic polynomial are smoother than the other method (using cubic polynomial). The effect is more pronounced for the period of transfer phase (time 0-1.26 s). The rest of time (when the leg on the ground), the effect is less, as the leg is in contact from both sides. It is worth to mention that the velocity of both tibia and ankle joints is equal to zero when the leg is in stance phase, due to there are no direct role for these joint in propelling the robot's body.



(a) Hip joint velocity.



(b) Tibia joint velocity.



(c) Ankle joint velocity.

Figure 3.6: The effect of the trajectory smoothing on the velocity of leg1 joints.

In torque controlling method, the produced torque at joints is also effected by the method of trajectory generation. Figures (3.7, 3.8, 3.9) depict a comparison between torques at leg 1 joints. The reference trajectories ,which created by cubic and quintic polynomial, have been exploited to produce joints torque.

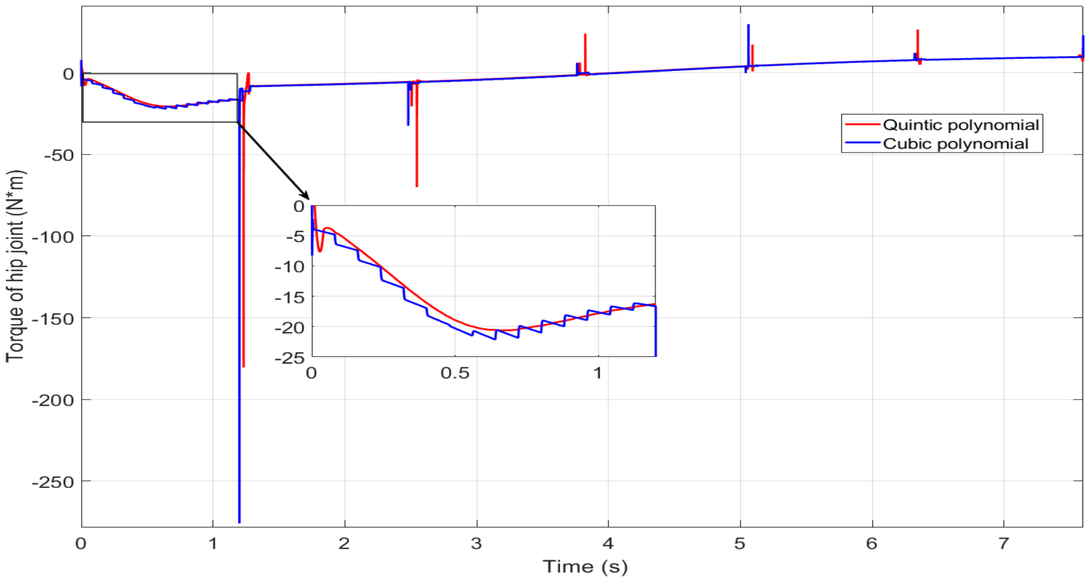


Figure 3.7: The torque at hip joint of leg 1, the blue and red curves represent the torque at hip joint, which produced by cubic and quintic polynomial respectively. Both signals at time 0-1.5 s have been magnified to indicate the difference between them.

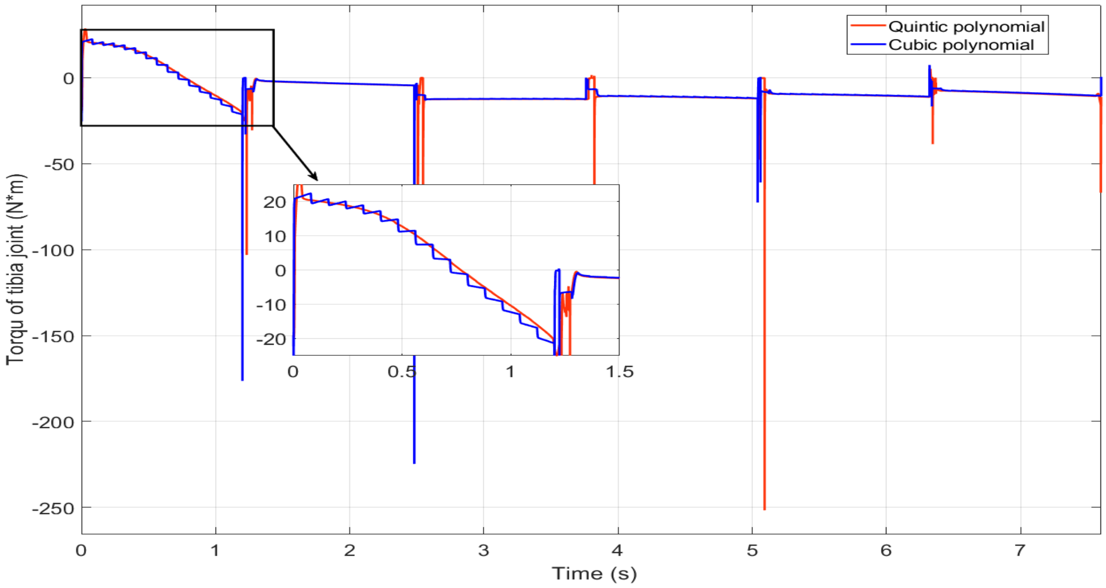


Figure 3.8: The torque at tibia joint of leg 1, the blue and red curves represent the torque at tibia joint, which produced by cubic and quintic polynomial respectively. Both signals at time 0-1.5 s have been magnified to indicate the difference between them.

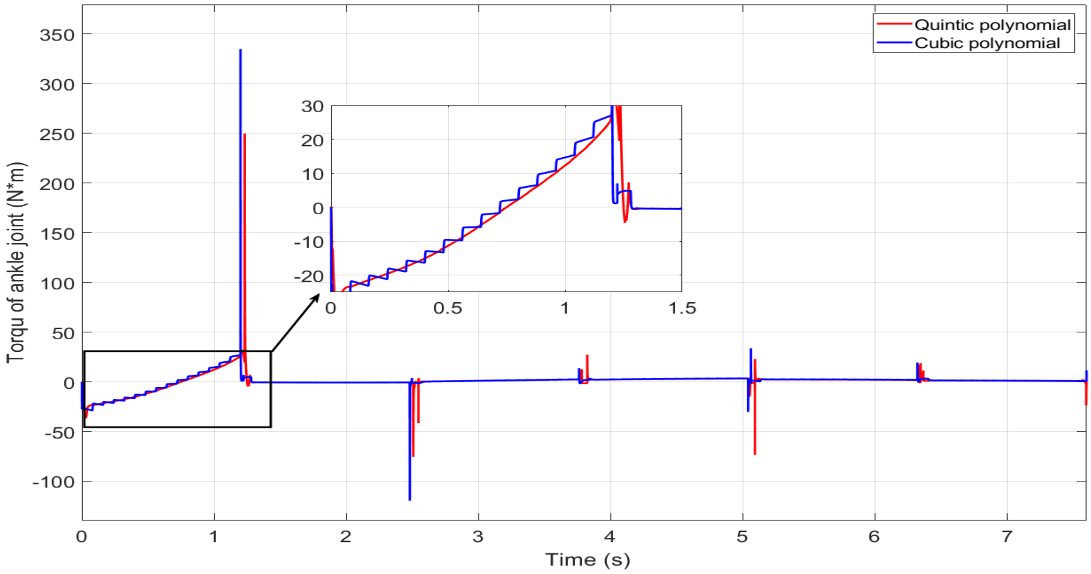


Figure 3.9: The torque at ankle joint of leg 1, the blue and red curves represent the torque at ankle joint, which produced by cubic and quintic polynomial respectively. Both signals at time 0-1.5 s have been magnified to indicate the difference between them.

A far from the configuration space, the localization of foot tip is also affected by the impact of trajectory state. Figure 3.10 illustrates the path of foot tip of leg 1 using cubic (blue line) and quintic (red line). In case of cubic polynomial and due to slippage, the position of foot tip along x-axis at stance phase is regressed. The amount of this regression will continue as long as the robot continues to move forward.

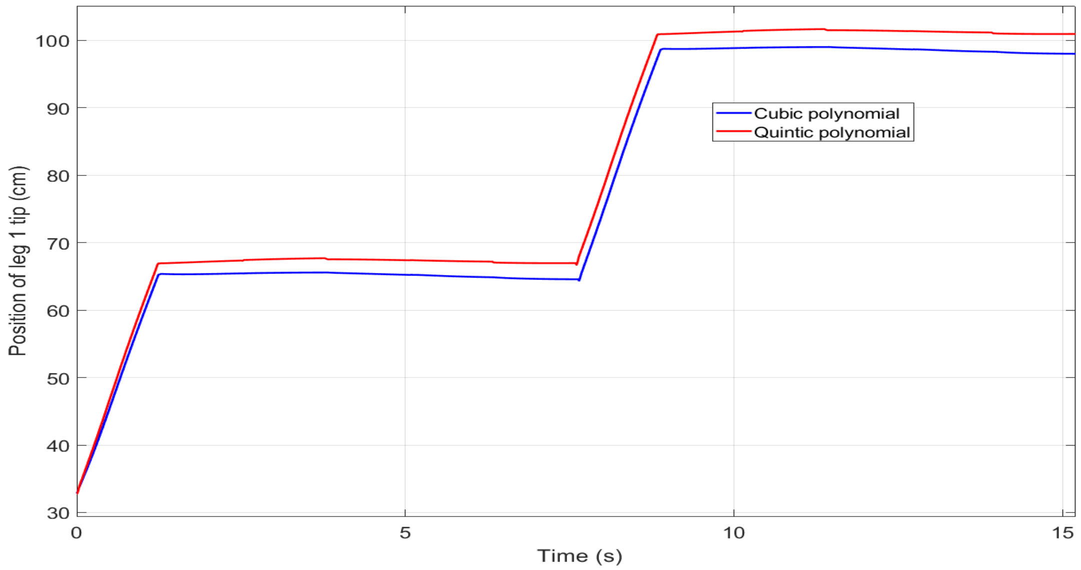
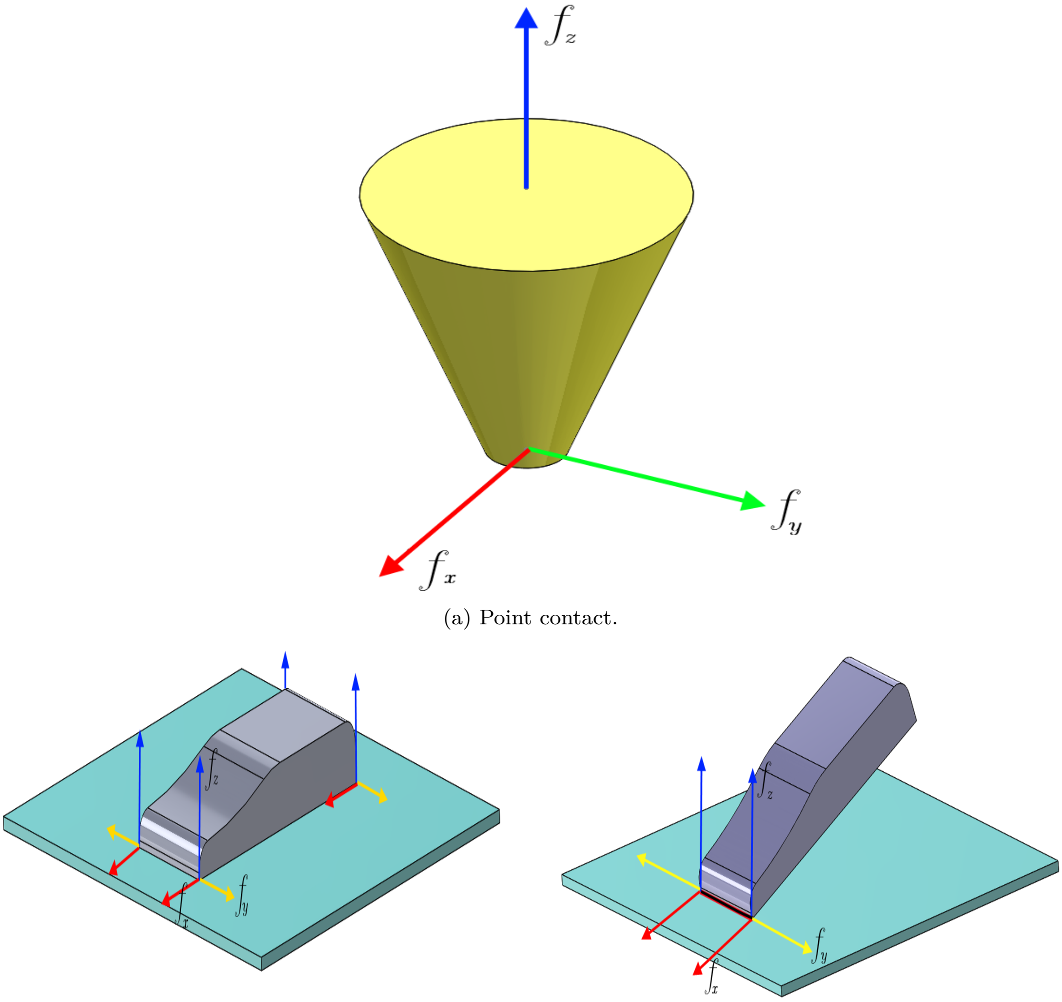


Figure 3.10: The effect of trajectory smoothing on the overall performance of the robot. This figure shows the foot tip trajectory along x-axis, The red line is a trajectory generated by cubic polynomial and the blue trajectory is generated using quintic polynomial.

3.6 The Contact Force in the Robot Dynamic

According to the structure of the legged robot, there are three types of rigid contact between the robot's legs and the environment [60]. These types are classified as point (in case of six-legged or four-legged robot) Figure 3.11a, line (edge contact) Figure 3.11c, and surface contact (in case of humanoid robot) Figure 3.11b. Each type of contact imposes a different rather constraints [26]. As the contact points increases, the number and type of the imposed constraints are raised accordingly [92].

While walking, the contact constraints will vary and the dynamics of the robot will be changed [139]. In general, modelling the contact by assuming the contact point as a fixed-base is an unattainable goal due to the switching transition and the probability of slippage [60]. Modelling all force components will cause complicating to the problem.



(a) Point contact.

(b) Plane Contact

(c) Edge Contact.

Figure 3.11: Types of contacts between the legged robot and the environment.

$$F_i = \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix}_{3\times 1} \tag{3.17}$$

where $F_i \in \Re^{1\times 3}$ is the net force at the contact i and $[f_{ix}, f_{iy}, f_{iz}]^\intercal$ are the forces in $(x, y, \text{and } z)$ directions respectively.

There are many approaches to handle the conundrum of force distribution in legged systems. Most of these approaches are based on optimisation methods [92, 140, 141, 142, 139]. Each point contact consists of three components as in Equation (3.17).

$$\mathit{\Gamma} = \begin{bmatrix} F_1^{\intercal} \\ F_2^{\intercal} \\ \vdots \\ F_n^{\intercal} \end{bmatrix}_{n\times 3} \tag{3.18}$$

According to the number of contacts, the stability equation can be written in a matrix form as in Equation (3.18). The full equilibrium equations are depicted in Appendix B [143].

$$\mathit{\Delta}\mathit{\Gamma} = F_{CoG} \tag{3.19}$$

where

$$\mathit{\Delta} = \begin{bmatrix} I_3 & I_3 & \cdots & I_3 \\ \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix} \tag{3.20}$$

$$\sigma = \begin{bmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{bmatrix} \tag{3.21}$$

$\mathit{\Delta} \in \Re^{6\times 18}$ in case of all legs of hexapod are on ground, is the matrix of coefficients, and $F_{CoG} \in \Re^{1\times 6}$ is the force and moment acting at the body CoG.

$$f_z \geq 0 \tag{3.22}$$

$$\sqrt{(f_x^2 + f_y^2)} \leq \gamma f_z \tag{3.23}$$

The effect of the contact force at foot on the CoG can be written as linear form, as in Equation (3.19). As it is clear from Equation (3.19) that the matrix $\varDelta$ is not square even with three legs in contact with ground, hence solving Equation (3.19) for $\varGamma$ is not straightforward. Two conditions should be satisfied, these constraints are: the normal force f_z should be greater than zero Equation (3.22), and $\sqrt{(f_x^2 + f_y^2)}$ should be less than the normal force multiplied by the static coefficient of friction γ, Equation (3.23).

There are two interesting approaches to remedy the problem of contact. In order to reduce the problem size, both of them considered the force and the moment at the CoG. Park et al. [60] extended the approach of [55] (weighting acceleration energy by the inertia matrix) to a torque minimisation problem, as below.

$$\min_{\tau \ni \Re^n} \frac{1}{2} \tau^\intercal W \tau \tag{3.24}$$

where

$$W = SM^{-1}(I - (J_c M^{-1} J_c^\intercal) J_c M^{-1}) S^\intercal \tag{3.25}$$

$S = [0_{6\times 6}\ I_{n\times n}]$, M is the inertia matrix, and J_c is the Jacobian of the contact point. The resulting τ according to this approach represent torques that dynamically consistent with contact forces.

$$J_c^\intercal = Q[R^\intercal\ 0]^\intercal \tag{3.26}$$

$$Q_c^\intercal (M\ddot{q} + H + G) = Q_c^\intercal S^\intercal \tau + R\varGamma \tag{3.27}$$

$$Q_u^\intercal (M\ddot{q} + H + G) = Q_u^\intercal S^\intercal \tau \tag{3.28}$$

where $Q \in \Re^{(n+6)\times(n+6)}$, $R \in \Re^{6\times 6}$ are orthogonal matrix, and upper triangular matrix respectively.

Righetti et al. [101] decomposed Equation (3.13) into constrained (3.27) and unconstrained (3.28) equations using *QR decomposition* method. Then, they formulated an minimisation problem using QP for unconstrained equation only.

In order to prevent legs from slippage, Equation (3.19) must be satisfied. However, dealing with non-linear constraint is very hard [144]. Therefore, converting the friction cone Figure 3.11a to pyramid inscribed inside the friction cone Figure 3.12 [142].

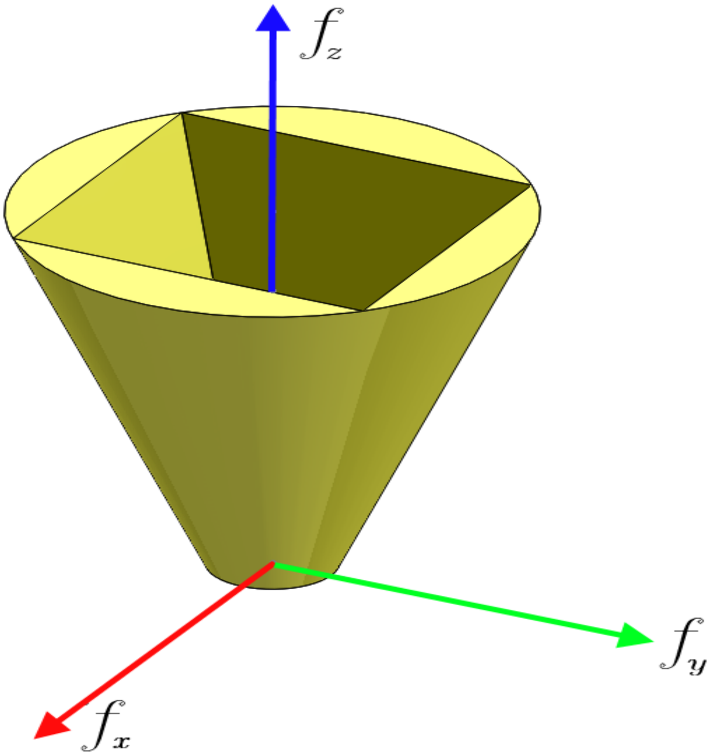


Figure 3.12: Friction cone around each foot-tip.

$$\begin{aligned}
-f_x + \frac{\gamma}{\sqrt{2}} f_z &\leq 0 \\
f_x + \frac{\gamma}{\sqrt{2}} f_z &\leq 0 \\
-f_y + \frac{\gamma}{\sqrt{2}} f_z &\leq 0 \\
f_y + \frac{\gamma}{\sqrt{2}} f_z &\leq 0
\end{aligned} \tag{3.29}$$

Equation (3.29) represents a set of simplified linear inequality conditions.

In this thesis, a virtual-links are assumed to define the relation between the robot and the ground [60]. This links consist of six joints with 6-DoF's three prismatic joints represent the transition in Cartesian space and three rotational joint represent the orientation. The CoG of the robot has been chosen as a connection point with world as illustrated in Figure 3.13.

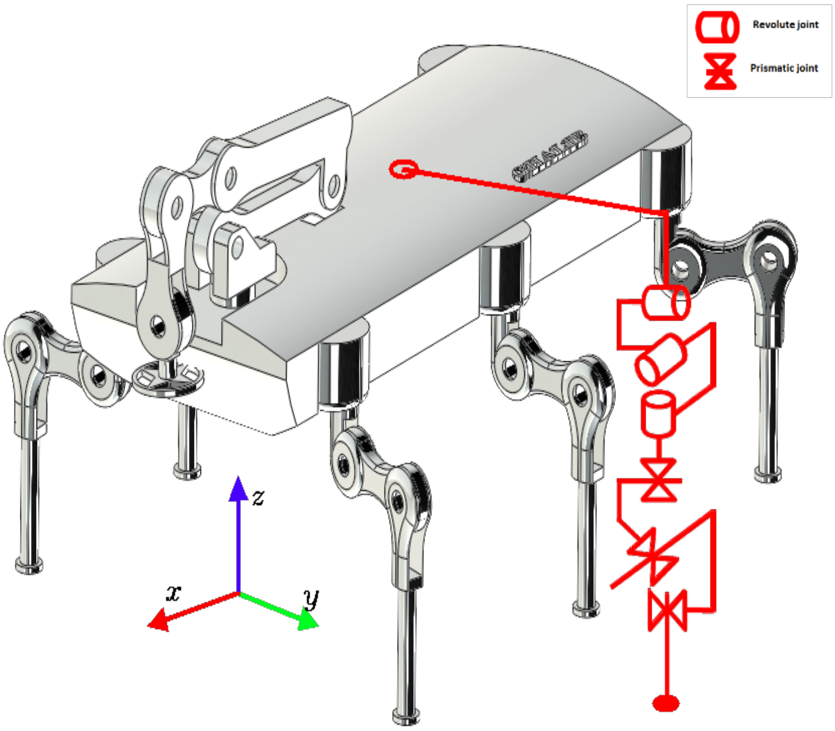
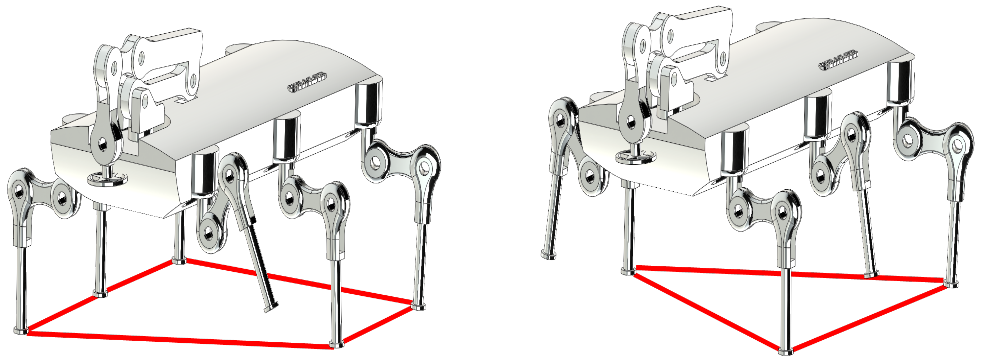


Figure 3.13: Virtual-links between the robot and the ground.

Six-legged robot has at least three legs in contact with ground (the case of tripod gait). The stance legs formed a polygon underneath the robot for each instant of time, Figure 3.14. Therefore, a single contact would be enough to describe the virtual connection between each foot and the contact surface. To ensure proper functionality, the contact point should be considered by a certain limit of the normal force and tangential forces Figure 3.11a.



(a) Five legs support.

(b) Three legs support.

Figure 3.14: Legs support types in six-legged robot.

Due to more than one leg in contact with ground, the effect of contact force at switching time from transfer phase to stance phase is remarkable in case of six-legged robot. Figure 3.15 shows the effect of normal force at legs (6, 3, 2, 5, and 4) on normal force at leg 1. The effect of internal force has been examined on the robot with gait sequence (1, 6, 3, 2, 5, 4) and duty factor of (5/6). The legs (1, 3, 5) are placed in left side and legs (2, 4, 6) are laid in other side. The top part of the figure illustrate the value of normal force for two states, namely, transfer phase (the green area at time 0-0.85s) and stance phase (the yellow area at 0.85 - 4.8s). The normal force equal 3850 N at time of contact, this big value is expected because of the impact of the contact. During stance phase, the normal force at leg 1 stay stable with very small value until time of leg 6 to switch from transfer to stance. The impact of the contact force at leg 6 causes a spike in the value of normal force at leg 1. Same effect would be happened at time of contact of leg 2 or leg 4, as both legs are placed in opposite side of leg 1. The effect would be reversed at time of contact of leg 3 or leg 5. This is expected as both legs are in same side of leg 1. The bottom part of figure 3.16 illustrates the normal force of legs (6, 3, 2, 5, and 4). The value of normal forces at different legs are varied depending on the leg position as shown in table 3.3. The influence of internal force can be attenuated by smoothing leg trajectory more as discussed in section 3.5. Smoothing trajectories using optimisation methods, such as genetic algorithm, would be an appropriate method to get such trajectory[145, 146]. However, trajectory smoothing is out of the scope of this thesis.

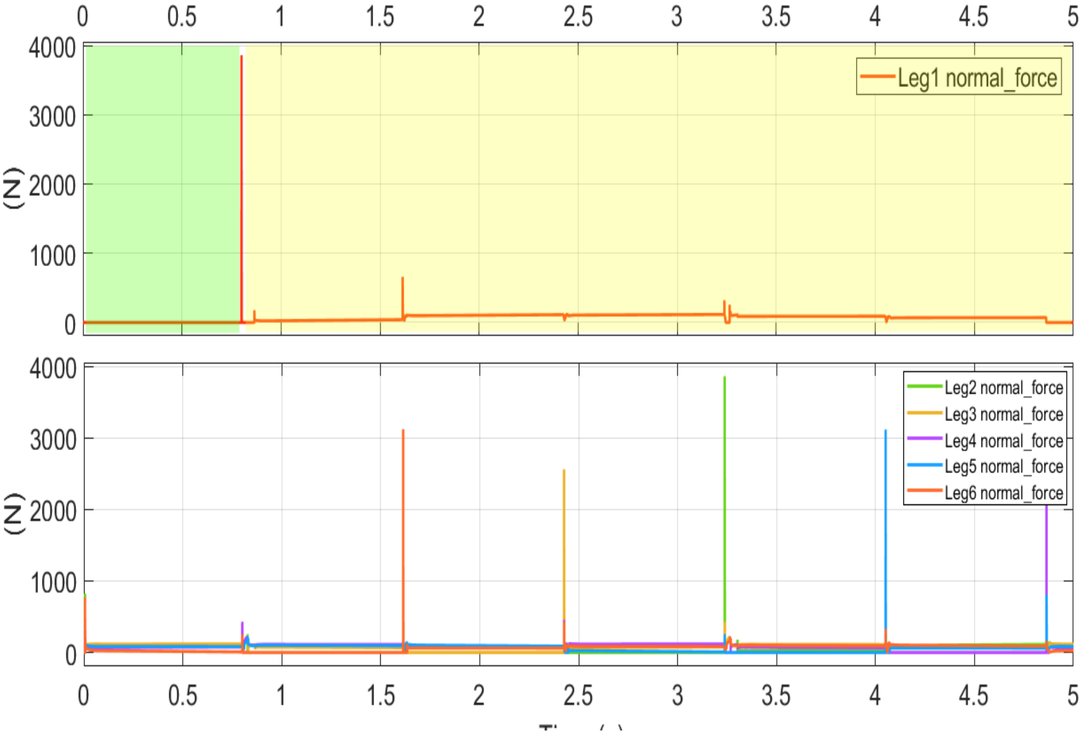


Figure 3.15: The green area at time 0 - 0.8 s represents the transfer phase of leg 1, the value of normal force is equal to zero, as no contact between robot and ground. The stance phase represented by yellow area at time 0.8 - 4.8 s, this portion of figure illustrates the impact of contact between leg 1 and ground, also it explains the effect of other legs on normal force at leg 1.

Table 3.3: The values of normal force at time of contact of legs.

Leg	Normal Force (N)
Leg 1	3859.7
Leg 2	3861.2
Leg 3	2560.3
Leg 4	2422.3
Leg 5	3118.4
Leg 6	3121.6

In terms of friction force and with same analogy, legs (2, 4, 6) have a greater influence on the value of friction force at leg 1 than legs (3 and 5). Figure 3.16 shows the impact of friction force of legs (6, 3, 2, 5, 4) on the friction force at leg 1. These measurements have been conducted on the robot with a gait sequence (1, 6, 3, 2, 5, 4) and with a duty factor of (5/6). Table 3.4 shows the values of friction force at each leg.

Table 3.4: The values of friction forces at time of contact.

Leg	Friction Force (N)
Leg 1	1441.9
Leg 2	1440.9
Leg 3	828.7
Leg 4	1453.4
Leg 5	920.4
Leg 6	925.7

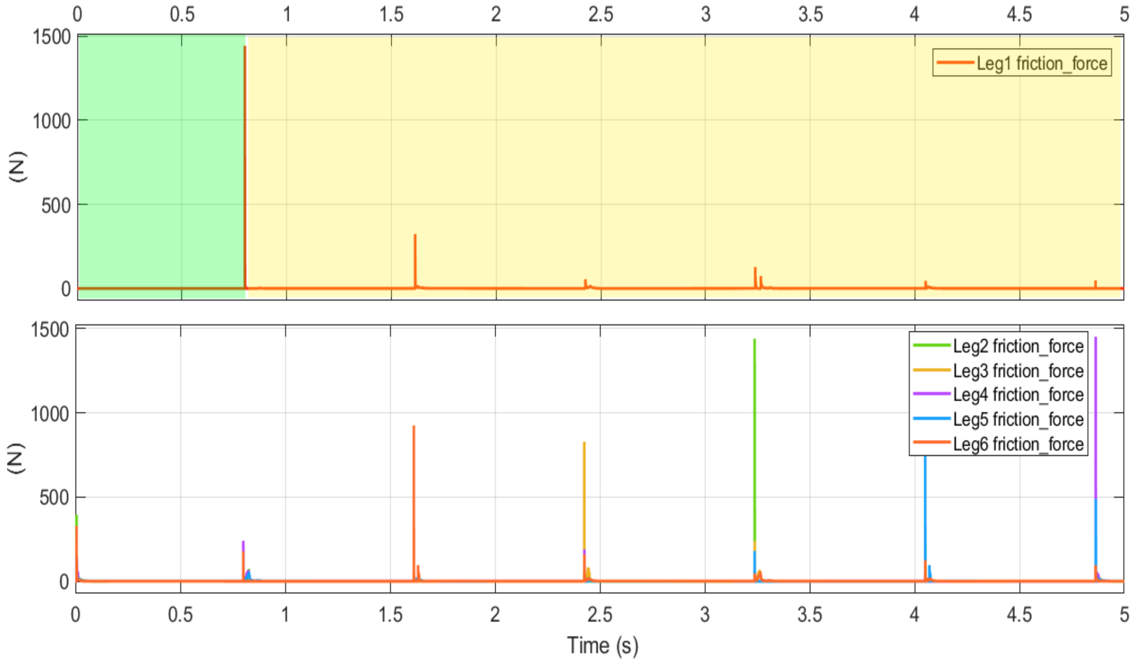


Figure 3.16: The top part of the figure demonstrates the friction force at leg 1. The green area at time 0 -0.8 s represents the transfer phase of leg 1, the value of friction force is equal to zero, as no contact between robot and ground. The stance phase represented by yellow area at time 0.8-4.8 s, this portion of figure illustrates the effect of other legs on friction force at leg 1.

3.7 Sensors

In order to perform the assigned tasks and to integrate its interaction with the environment, the robot is equipped with two types of sensors, namely, proprioceptive sensors and exteroceptive sensors. The first type gives the ability to sense the internal state of the robot, for instant, joint position, velocity, and torque sensors. The second type allows the robot interact with the surrounding. These sensors including vision, distance, and force sensors are used to detect the workspace of the robot.

In this section, brief information will be given about the used sensors. These sensors are distributed around the robot to measure current configuration. To do so, simscape provides different kind of sensor such as, position, velocity, acceleration, and torque for each joint. Furthermore, it gives the ability to sense the difference between different frames, which is useful to figure out the exact location of the robot with respect to the inertial frame. Definition of a number of reference frames connected to the robot on one side and the environment on other side are required in case of floating base robot. These frames represent virtual joints. The frames between foot tip of each leg and the manipulator end-effector with the world frame should be established.

Measuring the acceleration in x, y, z direction and the angular velocity around the CoM of the robot body are provided by three axis accelerometer and three axis gyroscope. Sensing the gravity can be achieved by inertial measurement unit (imu), which is measure the gravity independently from the environment conditions.

Measuring the force contact and friction at the contact point between the robot and ground is required to cope with interaction impact. For instance, when the robot walks over irregular terrain, the legs must absorb this impact by changing its configuration to keep the robot's body within a certain level, and keeping the friction force within specified limit will prevent from slippage.

Three cameras are used to figure out the presence of obstacles in front of the robot. One camera is installed underneath the robot body and attached to CoM frame. This camera provides a view along x-axis. The other two cameras are attached to right and left side of the arm end-effector respectively. Both these cameras give a view along y-axis. By integrating the view of all cameras, the path of the manipulator EE is generated.

3.8 Summary

The robot's platform was designed using SolidWroks; the CAD model of the robot has been exported to MATLAB/SimMechanics. The size of the considered robot platform and the robot's dimensions were explained in details.

A brief mathematical preliminary about the robot kinematics and dynamics modelling of the robot was introduced, and the types of control equation in velocity level and torque level have been presented. The requirements of operation space control were introduced. In order to recognise its difficulty, particularly in legged systems, a discussion about IK problem has been conducted. The floating-base characteristic together with redundancy and its impact on the control process have been argued.

Moreover, the trajectory generation and smoothing methods have been illustrated. The effect of trajectory smoothing on the overall behaviour of the robot was discussed. Two techniques have been considered, namely cubic polynomial and quantic polynomial. The effectiveness of both methods was tested on generated joint angles, joint velocity, joint torque, and legs slippage. In addition to its ability to produce smooth trajectory, the quantic polynomial method was exploited to create a continuous acceleration, which is required torque controlling methods.

As the robot has intrinsic interaction with the ground, the impact of the contact forces are in fact cannot be neglected. Therefore, modelling some physical effects, such as contact force and friction force in the contact points has considerable importance. The impacts of theses force on the performance of the robot are discussed, and the impact of the internal force, which produced by other legs, was explained.

Due to its flexibility to handle both linear equality and inequality equations, QP has been chosen to tackle with optimisation problem. Finally, some of used sensors were explored, such as, imu, gyroscope, encoders, and force sensors.

CHAPTER 4

Tracking Centre of Gravity Path

4.1 Introduction

The locomotion of a legged robot is very challenging mission; as several subtasks need to be performed at the same time [147]. These subtasks must be arranged according to their importance to the main task, for instance, transition from one pose to another could not be achieved without ensuring some necessary constraints (e.g., body balance). In case of legged robot, some constraints are imposed by the mechanic parts or dynamics of the robot, others are induced by interaction with the environments. Typically, the imposed constraints are formulated as a set of linear equality or inequality and stacked in a strict hierarchy according to their importance [148]. When designing the motion of the robot, a trade-off between the available motion space and the number of constraints that need to be satisfied should be considered [62]. In general, having large motion space (more DoF) enables the legged robots to execute more task and satisfy some constraints. However, the presence of passive joints at the CoG of the robot's body makes controlling the robot not direct [60].

In this chapter, the desired path of the robot is defined at its CoG and considered as a reference to track by the robot's legs. Each entity of this path includes the desired translation in Cartesian coordinates and orientation (Euler angles) of the robot. Although, the robot has plenty of DoF (18 DoF in the considered robot), tracking the reference path requires 6 DoF; therefore, there are 12 redundant DoF can be exploited to perform additional tasks. To this end, the challenge is how to calculate the value of each joint (18 DoF) by given a path with 6 DoF defined at CoG. Using the forward kinematics methods directly is not applicable due to the large number of DoF. Alternatively, task space control technique provides a proper approach to map the reference path from the operational space to the robot's configuration space using IK methods [55].

Utilising IK techniques to realise the robot motion are appealing [149]; however, matrix inversion involved in these techniques is always ill conditioned due to the number of unknowns are larger than the known variables. Therefore, solving this problem based on optimisation is more appropriate and provides a convenient way to get a minimum solution [94].

The problem of redundancy resolution at the inverse differential kinematics level is considered in this chapter to find the velocities in the configuration space, and the QP algorithm is used to handle the equality and inequality constraints in any level of priority.

In order to demonstrate the relation between the CoG of the robot and each foot tip, transformation matrix is derived using DH notation, and floating-base is represented using the virtual links.

The main contribution of this chapter is to formulate the inverse kinematics problem as an optimal problem and solved using QP solver. Following the idea of [150] to observe the joints limit and velocities bound, the major difference is that this approach are applied on six legged platform, which has more limbs to be controlled.

4.2 Kinematics Modelling of Six-Legged Robot

The DH notation is used to formulate the homogeneous transformation matrix between the CoG of the robot and the foot of each leg. This matrix describes the relation between frames in terms of translation in $x, y$and z coordinate and the orientation in roll, pitch, and yaw angles. The first transformation matrix is denoted as $T1$ which represents the transformation matrix between the CoG of the robot and the foot tip of leg1. In same way, $T2$, $T3$, $T4$, $T5$, and $T6$ represent the transformation matrices of leg2, leg3, leg4, leg5, and leg6 respectively with the CoG. In this section, the first matrix T1 will be derived; other matrices can be deduced similarly. Figure 4.1 shows the frame of CoG (f_{CoG}), the frame of body-hip joint (f_0), the frame of hip-tibia joint (f_1), the frame of tibia-ankle joint (f_2), and the frame at the contact (f_3).

Both f_{CoG} and f_0 share same rotation values around x, y and z-axis in different leg-bases. The position of f_0 for leg 1 related to f_{CoG} is (45, 24, 0) cm in x, y, and z respectively. The positions in Cartesian coordinate of hips of other legs are illustrated in Table 4.1.

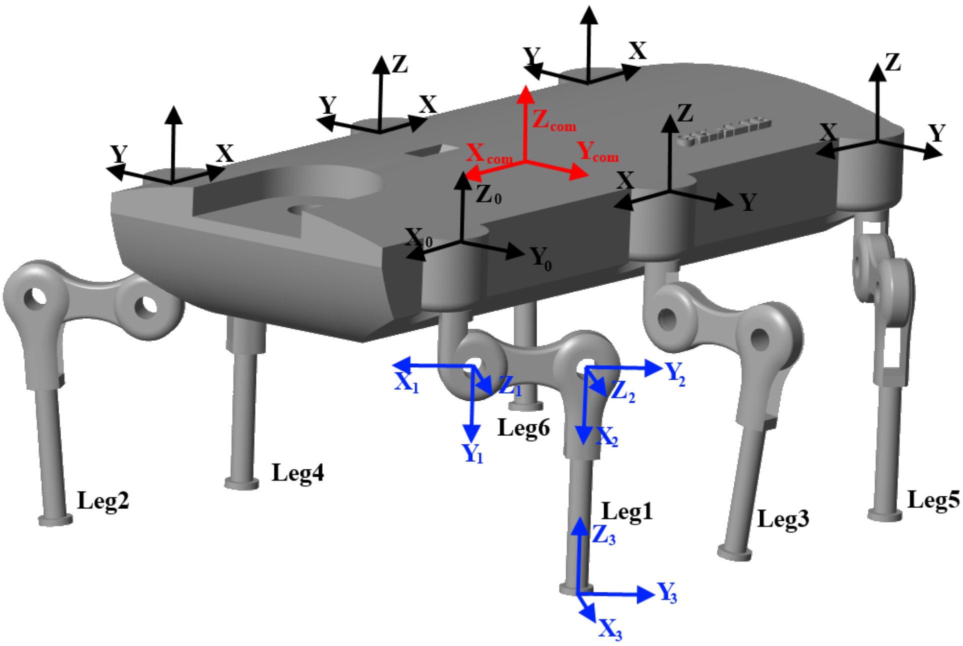


Figure 4.1: Frames assigned to body CoG, and all connection points between the robot's body and legs. The frames at leg one are shown to explain the position and the orientation of each joint in leg1.

Table 4.1: The position of hips of legs (2, 3, 4, 5, 6) with respect to body CoG.

Leg	position in x, y, z (cm)
Leg 2	(45, -24, 0)
Leg 3	(0, 24, 0)
Leg 4	(0, -24, 0)
Leg 5	(-45, 24, 0)
Leg 6	(-45, -24, 0)

While the leg (during transfer phase) has indirect influence on CoG, the translation and orientation of CoG are a function of translation and orientation of legs during stance phase. Therefore, the transformation matrix will be derived assuming the leg is in contact with ground. The spatial displacement between CoG and hip $T_{CoG,h}$, as the following:

$$T_{CoG,h} = \begin{bmatrix} 1 & 0 & 0 & 45 \\ 0 & 1 & 0 & 24 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

Other transformation matrices hip-tibia, tibia-ankle, and ankle-foot are denoted as $T_{h,t}$, $T_{t,a}$, and $T_{a,fo}$ are:

$$T_{h,t} = \begin{bmatrix} Cq_1 & 0 & Sq_1 & l_1Cq_1 \\ Sq_1 & 0 & -Cq_1 & l_1Cq_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$T_{t,a} = \begin{bmatrix} Cq_2 & -Sq_2 & 0 & l_2Cq_2 \\ Sq_2 & Cq_2 & 0 & l_2Sq_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$T_{a,fo} = \begin{bmatrix} Cq_3 & -Sq_3 & 0 & l_3Cq_3 \\ Sq_3 & Cq_3 & 0 & l_3Sq_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

where C and S are short for *cos* and *sin* respectively, l_1, l_2, and l_3 represent length of hip, tibia, ankle link respectively. The joints angle are indicated by q_1, q_2, and q_3.

$$T1 = \begin{bmatrix} Cq_1 & -Sq_1Cq_1 & Sq_1Cq_1 & p_x \\ Sq_1C(q_2+q_3) & -Cq_1S(q_2+q_3) & Cq_1 & p_y \\ S(q_2+q_3) & C(q_2+q_3) & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} (l_1 + l_2Cq_2 + l_3C(q_2+q_3))Cq_1 \\ (l_1 + l_2Cq_2 + l_3C(q_2+q_3))Sq_1 \\ l_2Sq_2 + l_3S(q_2+q_3) \end{bmatrix} \tag{4.6}$$

The overall transformation matrix between body CoG and foot-tip is as Equation (4.5). The position of the foot tip (p_x, p_y, p_z) can be represented with respect to leg frame as shown in Equation (4.6). The relation of the foot tip configuration vector $\boldsymbol{x}$ and the angles of a leg in the joint-space $\boldsymbol{q}$ represented as follows:

$$\boldsymbol{x} = f(\boldsymbol{q}) \tag{4.7}$$

where $x \in \Re^3$ is the position of the foot tip.

4.3 Mathematical Formulation

4.3.1 Floating-Base Representation

In general, a floating-base robot is, by definition, not fixed firmly to any location in the ground, controlling such a robot is still challenging because no actuation are implemented in its CoG [50]. In the following text, the fundamental notations and the Jacobian matrices required for floating-base are introduced. In addition to the actuated joint angles, the robot has 6 un-actuated joints, which represent the virtual link between the world frame and the CoG frame; the whole motion space can be expressed as below:

$$q = \begin{bmatrix} q_a & q_u \end{bmatrix} \tag{4.8}$$

where $q \in \Re^{1\times 24}$ are the whole joint angles, $q_a \in \Re^{1\times 18}$ are the joint of all legs, and $q_u \in \Re^{1\times 6}$ are the passive joints, which represent the position and orientation of the CoG.

$$J = \begin{bmatrix} J_a & J_u \end{bmatrix} \tag{4.9}$$

where J, J_a and J_u are the whole system Jacobian, robot joint Jacobian and robot base Jacobian respectively.

$$J_a = \begin{bmatrix} J_{leg1} & J_{leg2} & J_{leg3} & J_{leg4} & J_{leg5} & J_{leg6} \end{bmatrix} \tag{4.10}$$

where J_{leg1-6} represent the Jacobian of leg 1 - 6.

In the same manner, the Jacobian matrix of the whole system can be represented as Equation (4.9). Since all actuated joints exist in legs, the Jacobian of actuated joint can represented by Equation (4.10).

The control points, which are described by these Jacobian are the feet-tips, and any point on the robot body can be defined in the same way as a function of the whole body motion. The Jacobian that represented in Equation (4.9) can map the differential form of Equation (4.8) to the space of the tasks to be performed as:

$$\dot{\boldsymbol{x}} = J\dot{\boldsymbol{q}} \tag{4.11}$$

$$\dot{\boldsymbol{q}} = J^{-1}\dot{\boldsymbol{x}} \tag{4.12}$$

The dimension of $\dot{\boldsymbol{x}}$ can be determined by the number of tasks to be performed, this can be achieved by precisely define tasks. Inversion of Equation(4.11) is needed when the motion rate $\dot{\boldsymbol{q}}$ is required for a given tasks, as in Equation (4.12).

4.3.2 Controlling of Multiple Tasks

Inverse kinematics problems include defining the motion of the robot in task space then mapping the task reference to the configuration space according to Equation (4.12).

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{bmatrix} \tag{4.13}$$

where p_i represents the position and the orientation of the task points and k is the number of task points.

$$J = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_k \end{bmatrix} \tag{4.14}$$

In order to accomplish multiple tasks at the same time, the tasks are aggregated in one matrix [20], as shown in Equation (4.13). The corresponding Jacobian for several tasks is defined in Equation (4.14).

The overall behaviour of the robot is produced by controlling each task to achieve a particular object. However, solving Equation (4.12) cannot ensure a proper decoupling between tasks. The null space projection method has been used widely to solve the decoupling problem [80].

$$\dot{q} = J\dot{x}_1 + (I - J^{-1}J)\dot{x}_2 \tag{4.15}$$

where $(I - J^{-1}J)$ is a projector in the null-space of J.

Considering two tasks with different priority levels. The importance of task one ($\dot{x}_1$) is higher than task two ($\dot{x}_2$). In that case, projection task two in the null space of task one will ensure an appropriate decoupling between two tasks as in Equation (4.15).

While this approach is attractive to impose a strict decoupling between tasks, it cannot handle constraints formulated as inequality. Therefore, a good solver should be characterised by the ability to prevent the tasks from interference and can handle both equality and inequality constraints in any priority level [62].

4.4 Problem Formulation

Since Jacobian matrix is not square, *i.e.*, the number of rows (n) is greater than the number of columns (m), inverting it is not straight forward. Hence, solving equation (4.12) imposes formulating it in least-square form [86], and the minimum norm can be obtained using pseudo-inverse method or any generalised inverse techniques [96]. However, since the problem includes inequality constraints, such as joints-limit; the analytical solution cannot guarantee considering inequality constraints directly. Alternatively, potential field method has been used to get an approximate solution [151]. The prioritisation levels can be forced by projecting the lower priority tasks in the null-space of the higher one [82, 100].

In order to consider any type of linear constraints, QP algorithm has been widely used in legged-robot control to solve IK problems [87] and to solve Inverse Dynamics (ID) problem [101]. While the optimisation parameter in IK is the robot velocity, the robot torque is the optimisation parameter in ID.

$$\min_{\dot{q}\in\Re^n} \frac{1}{2}\,\dot{q}^T W \dot{q} \tag{4.16}$$

$$s.t. \quad \dot{x}_e = J_e \dot{q} \tag{4.17}$$

$$\dot{x}_i \leq J_i \dot{q} \tag{4.18}$$

where W is a weighted matrix (the weight of each joint velocity in each task), the subscript e and i denote to equality and inequality respectively.

Unlike the classical algorithms, such as pseudo-inverse and projection , the QP algorithms can intrinsically handle both types of constraints. Equations (4.16), (4.17), (4.18) represent a classical notation of QP algorithm with two sets of linear-equality and linear-inequality constraints.

The objective function is the velocity in joint space $\dot{q}$ subject to the error minimisation between velocity in task space $\dot{x}$ and velocity in joint-space $\dot{q}$.

$$J(q)\dot{q} - \dot{x} = 0 \tag{4.19}$$

During achieving tasks, controlling the robot encounters many constrains to be fulfilled. Generally, these constraints can be written as equality for instance, the velocity at a contact-point equal to zero, as in Equation (4.19).

$$q^{l_o} \leq q \leq q^{u_p} \tag{4.20}$$

$$\dot{q}^{l_o} \leq \dot{q} \leq \dot{q}^{u_p} \tag{4.21}$$

where l_o lower bound and u_p is the upper bound for both joints position and joints velocity. Some constraints can be formulated as inequality, such as, joints position and joints velocity within certain maximum and minimum limits, as shown in Equations (4.20) and (4.21) respectively.

The joint limit of hip, tibia, and ankle for each leg are summarised in Table 4.2.

Table 4.2: Joints limits of hip, tibia, and ankle for each leg.

Joint	Joints limits (degree)
Hip	(-35, 35)
Tibia	(-10, 30)
Ankle	(-10, 20)

Although the classical QP can handle both type of constraints, imposing strict hierarchy between tasks is elusive. An interesting method proposed by [87] to prioritise both equality and inequality linear equations in same optimisation level. This method is based on defining a set of feasible linear equality and inequity equations in least-square sense. Assuming $\mathit{\Lambda}, \Upsilon$ are two matrices in $\Re^{m \times n}$ and δ, ϵ are two vectors in $\Re^m$, as the following:

$$\min_{\xi \in \Re^n} \frac{1}{2} \parallel \mathit{\Lambda}\xi - \delta \parallel^2 \tag{4.22}$$

$$\begin{aligned} &\min_{\mu \in \Re^n} \frac{1}{2} \parallel \mu \parallel^2 \\ &s.t. \quad \Upsilon \xi - \epsilon \leq \mu \end{aligned} \tag{4.23}$$

where $\mu \in \Re^m$ is a vector of a slack variables.

$$\begin{aligned} &\min_{\xi \in \Re^n, \mu \in \Re^m} \frac{1}{2} \parallel \mathit{\Lambda} \xi - \delta \parallel^2 + \parallel \mu \parallel^2 \\ &s.t. \quad \Upsilon \xi - \epsilon \leq \mu \end{aligned} \tag{4.24}$$

According to the method, Equations (4.22), (4.23) were merged in one minimisation problem as in Equation (4.24).

$$S_{i+1} \quad = \quad \min_{\xi \in S_i, \mu \in \Re^m} \frac{1}{2} \parallel \mathit{\Lambda}_i \xi - \delta_i \parallel^2 + \parallel \mu \parallel^2 \tag{4.25}$$

$$s.t. \quad \Upsilon_i \xi - \epsilon_i \leq \mu \tag{4.26}$$

For i number of priority levels the set of next feasible solutions are induced from higher priority level as shown in Equations (4.25) and (4.26).

$$\min_{\xi, \mu_{i+1}} \frac{1}{2} \parallel \mu_{i+1} \parallel^2 \tag{4.27}$$

$$\begin{aligned} &s.t. \quad \Upsilon_i \xi - \epsilon_i \leq \mu_i^* \\ &\Upsilon_{i+1} \xi - \epsilon_{i+} \leq \mu_{i+1} \end{aligned} \tag{4.28}$$

To reduce the computational time, the above method was extended [150] to include the optimal solution from the previous level as a constraint when solved for next priority level as in Equations (4.27) and (4.28).

$$\varLambda = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{4.29}$$

To speed up the computation time [88] proposed a method relied on QR factorisation. The matrix $\varLambda$ is factorised to two matrices, orthogonal $Q \in \Re^{n\times n}$ and upper triangular $R \in \Re^{m\times m}$, as in Equation (4.29).

$$\xi = \xi_1 + V_1 v_1, v_1 \in \Re^{n-m} \tag{4.30}$$

For the linear equation $\varLambda_1 \xi = \delta_1$ with size m_1 and assuming $\varLambda_1$ is full row rank, this matrix can be factorised to Q_1 and R_1. The orthogonal can be split to $U_1 \in \Re^{n\times m_1}$, which represent the rang space of $\varLambda_1$ and $V_1 \in \Re^{n\times(n-m_1)}$, which represent the null-space of $\varLambda_1$. By matching with Equation (4.15), the general solution of δ is represented as Equation (4.30).

$$\min_{v_1 \in \Re^{n-m_1}} \frac{1}{2} \parallel \varLambda_2 V_1 v_1 - (\delta_2 - \varLambda_2 \xi_1) \parallel^2 \tag{4.31}$$

For the lower priority set of equations $\varLambda_2 \xi = \delta_2$ and by substituting the value of ξ from Equation (4.30) , the minimisation problem for second layer can be written as in Equation (4.31).

$$\begin{aligned} \min_{v_{i-1}} \frac{1}{2} \parallel \varLambda_i V_{i-1} v_{i-1} - (\delta_i - \varLambda_i \xi_{i-1}) \parallel^2 \\ s.t. \quad \Upsilon_i \xi \leq \epsilon_i \end{aligned} \tag{4.32}$$

Using this result for i number of tasks with linear inequality, Equation (4.24), the result can be written in general form, as in Equation (4.32).

4.5 Trajectory Generation

4.5.1 CoG Trajectory Generation

The fundamental method to address the static stability criteria is to examine the polygon that formed by the stance legs [24]. As long as the vertical projection of the robot's CoG lies inside the supported polygon the robot is statically stable. In static walk, at any moment of time, either 3 or 6 feet are in contact with the ground and act as stance legs. The robot's path is defined at its CoG, and formed from several points as:

$$T = \begin{bmatrix} R_o & p_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{4.33}$$

where $R_o \in \Re^{3\times3}$ is the rotation matrix of a certain point with respect to arm-base and p is a translational vector, which describe the path point.

The position and orientation of each point is described a transformation matrix as in Equation (4.33) with respect to the world frame. The legs in both the stance and transfer phases will track these points. In other words, the controller will create joint angles for all legs. By making the CoG track the assumed path, the stability of the body in moving forward is assured. The position of the CoG is completely controlled by the legs in stance phase.

4.5.2 Swing Leg Trajectory

The swing leg is guided to its target with the trajectory created by fifth order spline. The coordinate system is considered so that x-axis represents the forward direction (longitudinal axis) and y-axis indicate to the transverse axis. The legs operate in two different phases: swing phase and stance phase. In the swing phase the leg lifts up and transferred to the next foothold. In the second phase, the leg stays on the surface and propel the body forward.

$$q\,(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{4.34}$$

where a_j , $j = 1, 2, 3, 4, 5$ are the coefficient, whose value are determined using a set of boundary conditions defined over the transfer phase for each joint. The trajectory of leg in swing phase assumed to follow a polynomial of fifth order as in Equation (4.34). The boundary conditions of joint angles, velocities at initial and final points of the trajectory are applied to determine six coefficients for the trajectory of each joint. Figure 4.2 explains the path of leg one foot at transfer phase for two situations.

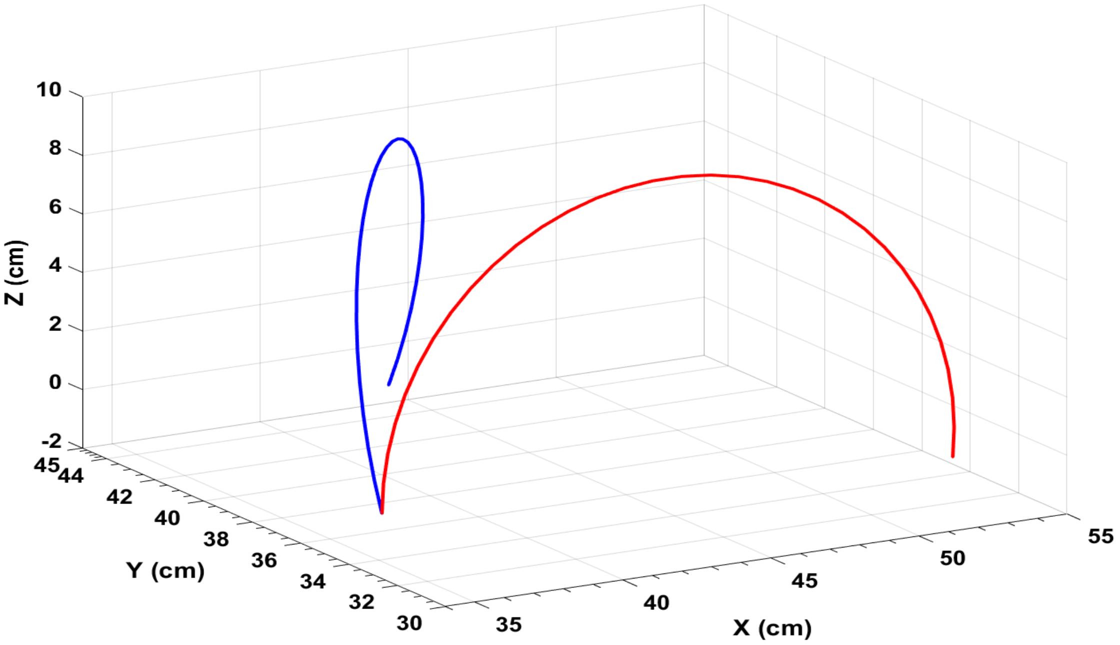


Figure 4.2: Leg 1 foot path at transfer phase. The red semi-circle represents the path of leg 1 in case of straight line. Case of turning to the left side by 10%, the foot path is represented by blue curve.

4.6 Experiments

In order to verify the effectiveness of the motion generation algorithm, three experiments have been done. Set of initial parameters are given before starting the evaluation: linear velocities to determine the robot's direction, the angular velocities at CoG to specify the orientation of the robot, the initial position, and final position of the overall path. The legs transfer sequence is set up as (leg1, leg6, leg3, leg2, leg5, and leg4) to get a proper level of balance. First experiment, the bath of the robot is defined as straight-line starting from the origin and ending at point 30 cm along x-axis. Second experiment, in order to demonstrate the ability of the robot to navigate in a different mode, the path is determined along y-axis. Third experiment, a turning to left at CoG around z-axis by $10°$ is added to whole path.

4.6.1 Tracking a straight line along x-axis at CoG

In this experiment, the translation profile of the robot is defined by starting point at origin with vertical distance 30 cm and the ending point at (30, 0, and 30) cm. In addition; the rotation angle is set to zero. The duty factor is equal to (5/6). To ensure this distance is achieved by the robot in one cycle, the stride length is set to (endpoint/2). Figure 4.3 shows the robot follows a straight line. The robot bath is indicated by black line in x, y plane. The starting and ending point are demonstrated by yellow and green ball. The length of red line at CoG represents the translation distance along x-axis. To demonstrate the translation and rotation relations between legs and the robot's CoG, all legs bases are indicated by green lines. The blue and red semi-circles represent the path of robot's feet.

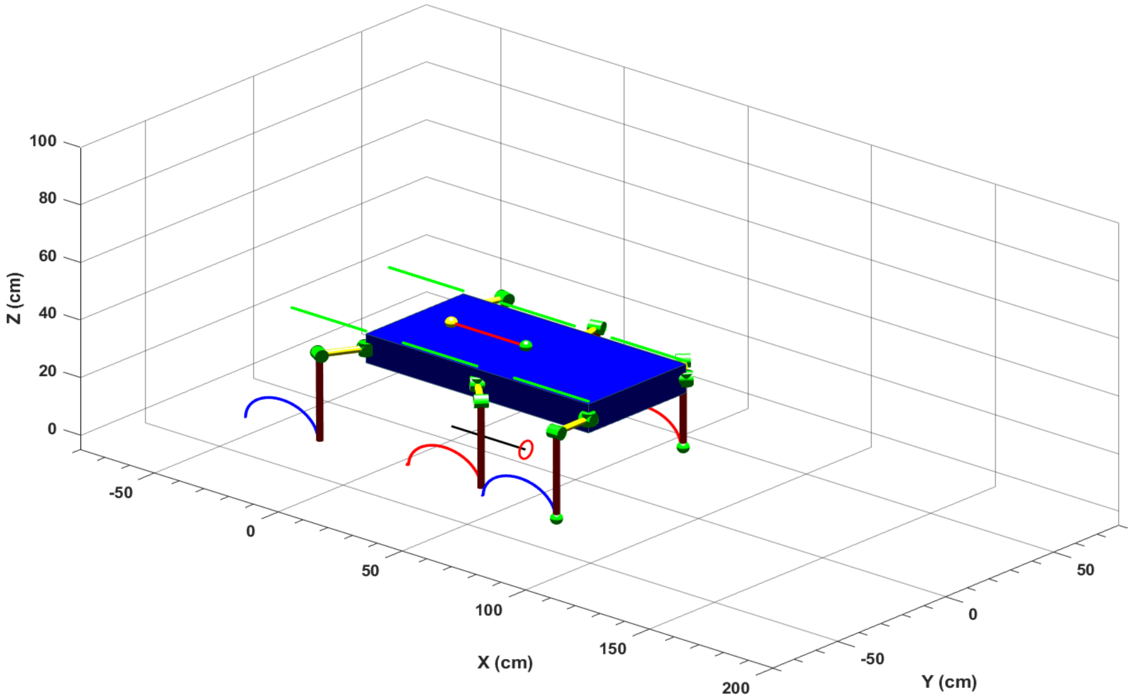


Figure 4.3: The robot follows a straight line with 30 cm length along x-axis. The red line represents the overall path of the robot body. The starting and ending points are represented by yellow and green balls. The blue and red semi-circles indicate the path of the robot's feet.

Figure 4.4 shows the trajectory of leg one joints for one cycle. Time of whole cycle is (7.6 s). This is divided to (1/6) as transfer time and (5/6) as stance time for one leg. According to the leg situation and stride length, two constrains have been imposed. At transfer part, the rang of all legs joint are limited to (-1, 1) rad for hip joint, (0, 0.8) rad for tibia joint, and (-0.8, 0.2) rad for ankle joint. The initial values of leg one joints are (0.785, 0, 0) rad for hip, tibia, and ankle respectively, these values give leg one a configuration to start transfer phase. The joint trajectory is generated according to the base position and orientation, which is a function of CoG position and orientation. Figure 4.5 illustrates the joints angles of other legs (leg6, leg3, leg2, leg5, and leg4). The figure shows the different phases between legs to achieve the overall motion of the robot.

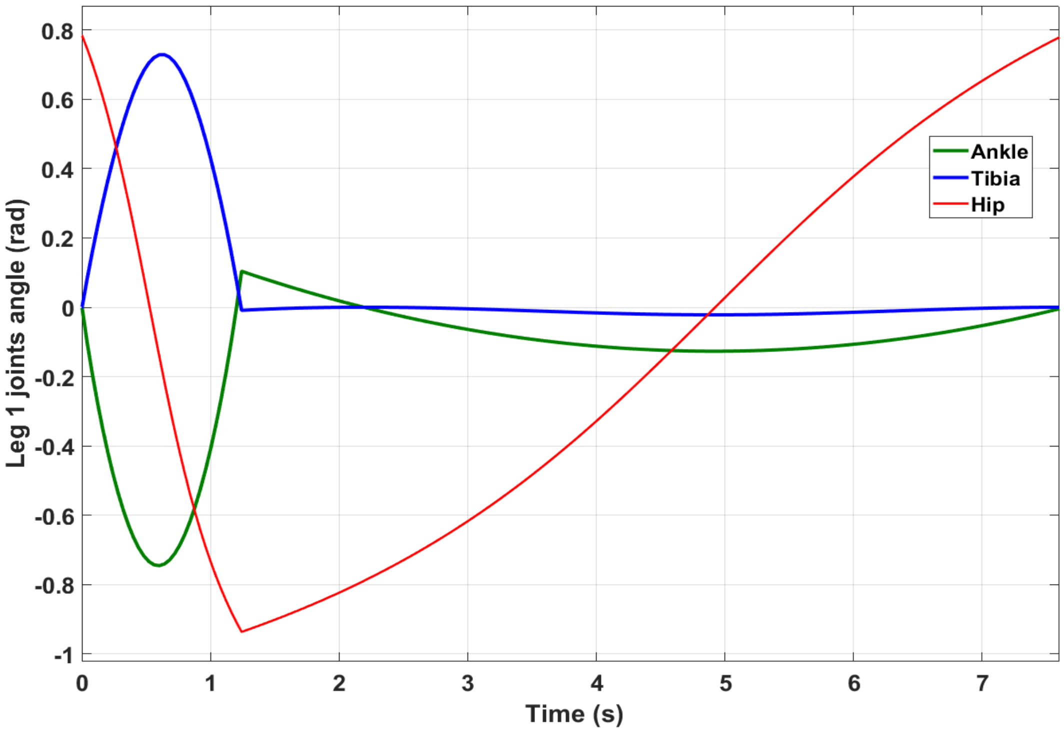
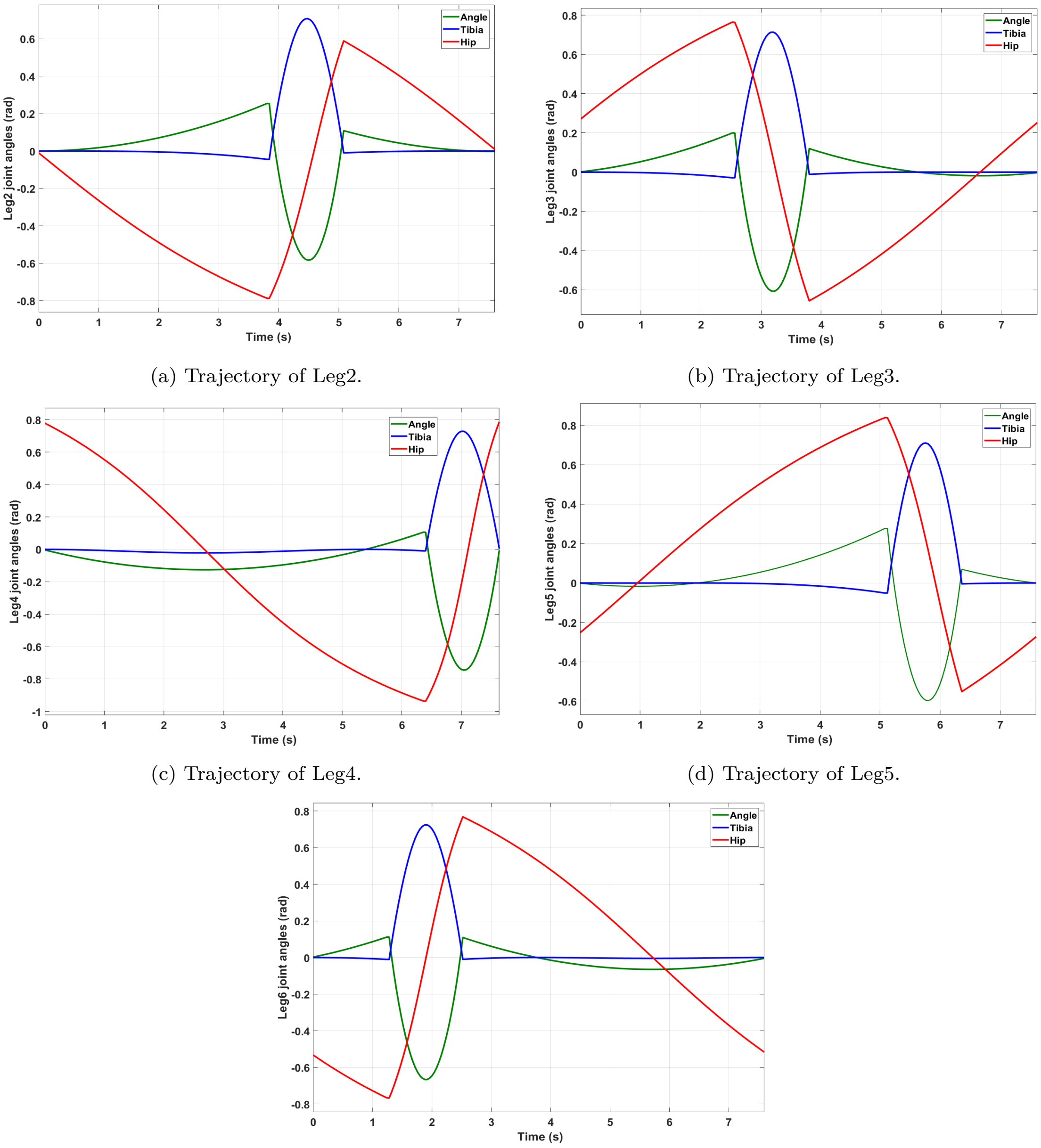


Figure 4.4: Trajectory of leg 1 joints for one cycle. The leg starts with transfer phase between time (0 - 1.3) s. During this period, the hip joint moved the leg from back to forth, and the tibia joint has changed the leg configuration from down to up and then up to down with time equal (transfer phase time/2) for each. The leg start stance landing at time 1.3 s until time 7.6 s.



(a) Trajectory of Leg2.

(b) Trajectory of Leg3.

(c) Trajectory of Leg4.

(d) Trajectory of Leg5.

(e) Trajectory of Leg6.

Figure 4.5: Trajectory of robot legs, case of tracking straight line along x-axis at CoG.

The contact point between leg and the ground is represented by a passive joint. Therefore, in order to keep firm liaison between legs and ground, the velocity at the contact point should be equal to zero. Figure 4.6 shows the position of leg one foot with respect to x-axis. At transfer phase the foot tip moved from point 34 cm to point 66 cm *i.e.* a distance of 32 cm. This distance is achieved at time 0-1.3 s. There is a slight change (0.24 cm) in the position of foot-tip during time 1.3 - 7.6 s, this tiny slippage due to the rotation of the foot-tip.

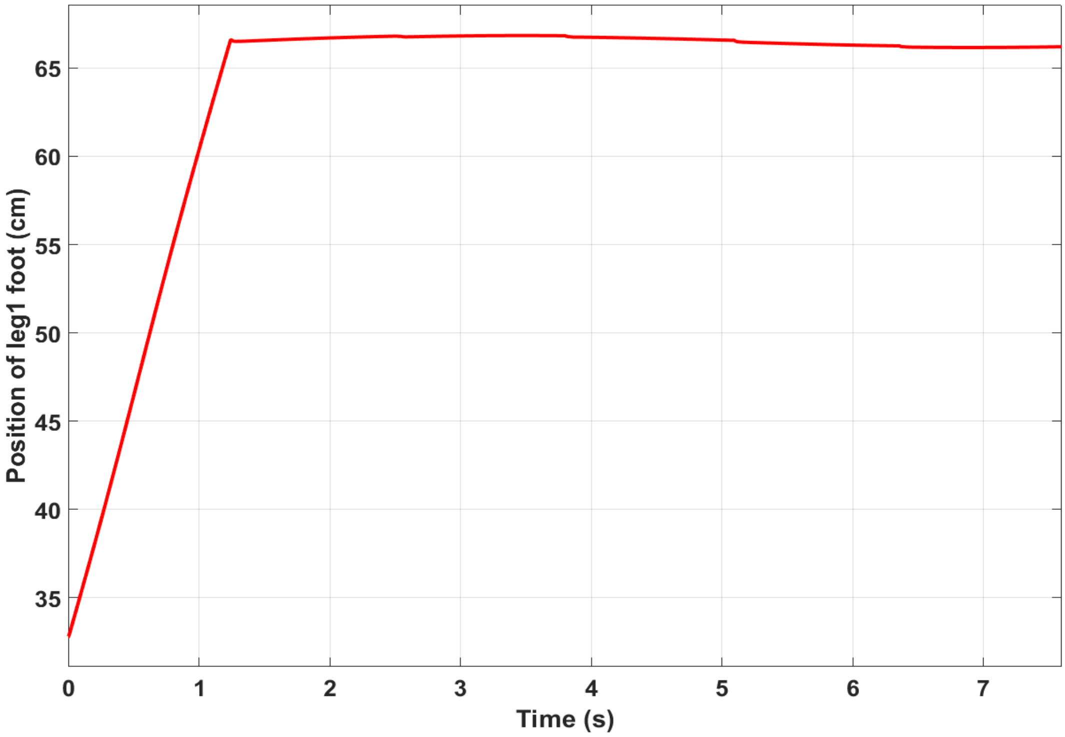
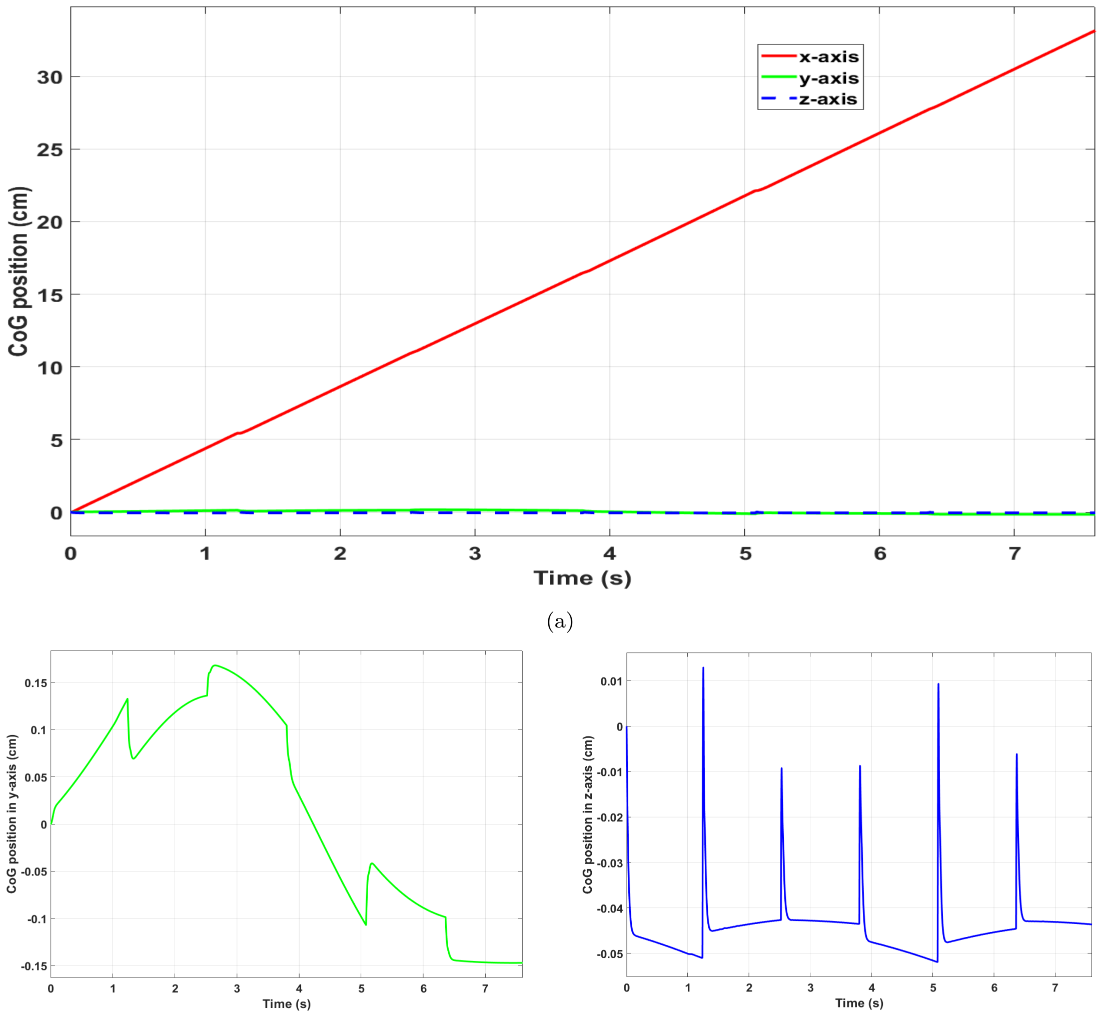


Figure 4.6: Location of leg's one foot along x-axis. While, the foot tip is moved a distance of 32 cm during transfer phase, it is standstill at the stance pace time. The deceleration before landing will help to decrease the impact of colliding.

Figure 4.7 illustrates the transition of robot's CoG in 7.8 s. The CoG moved from 0-32 cm along x-axis. Although the reference path is set to 0 in y direction, the actual path of the robot is swung around the reference by 2 cm around the reference path. However, this small value of deviation does not have much effect on the overall balance if it is compared with the width of the robot's body 85 cm. The path in z direction remained constance for the whole cycle, except for very short time with very small value at time of contact.



(a)

(b)

(c)

Figure 4.7: Position of CoG in x, y, and z direction. The top figure illustrate the Cartesian coordinates of the body CoG. The red line depict the distance that the robot was travelled in 7.8 s. The deviation from y-axis is represented by green curve in figure b. Figure c shows the real path in z-axis.

The robot walks at a constant velocity is about 0.044 m/s. Figure 4.8 demonstrates the velocity of the robot in x-axis. Although, there is a drop for very short time in the value of the robot velocity at the time of contact, the overall speed of the body has no zero values. Excluding the spikes at the contact time, the velocity in y direction remain constant with 0 m/s as shown in Figure 4.9.

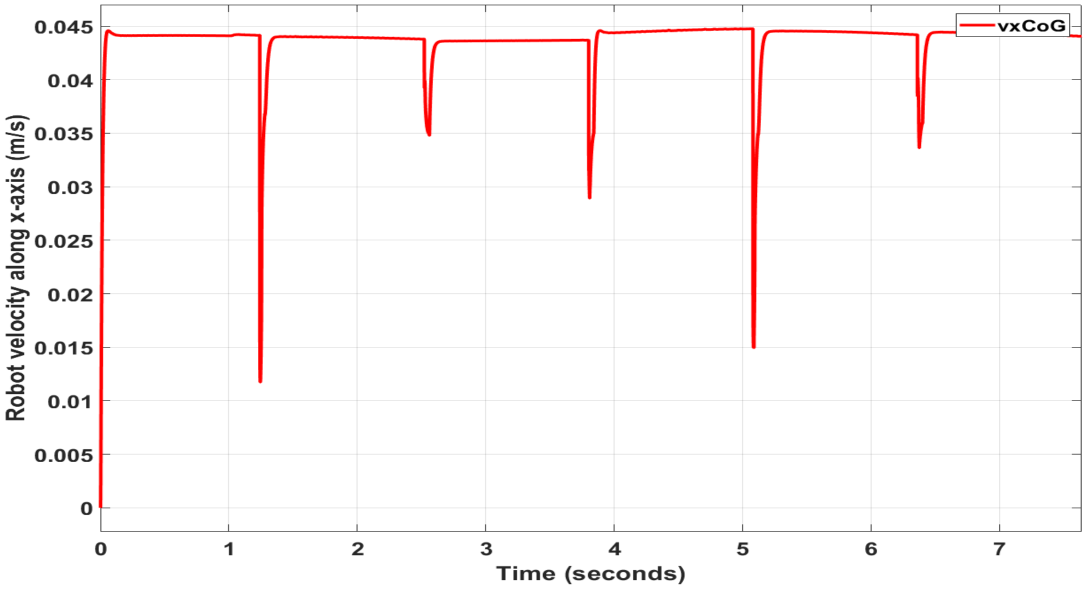


Figure 4.8: Velocity of CoG in x direction. The robot needs 0.044 m/s speed to get 32 cm in 7.6 s.

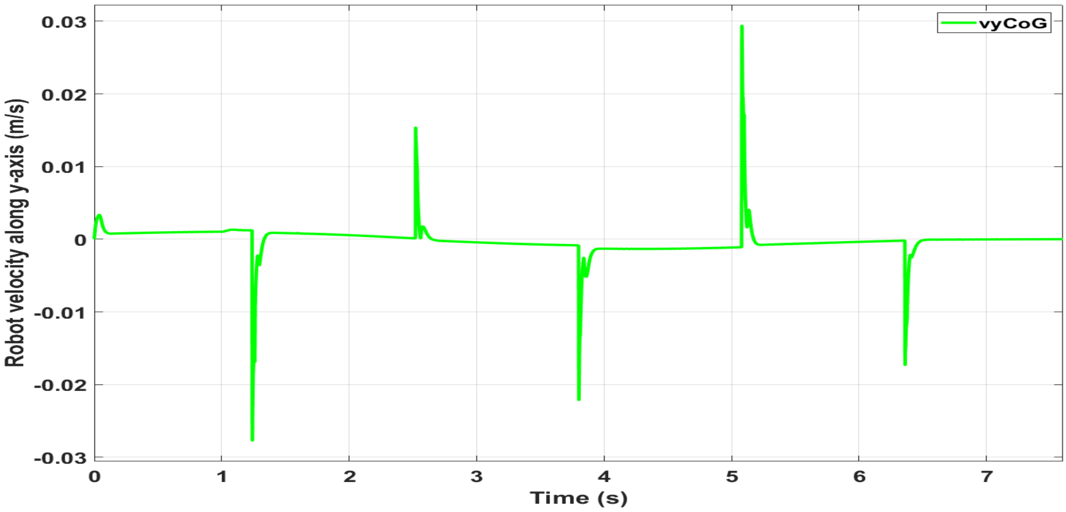


Figure 4.9: Velocity of CoG in y direction. The velocity in y direction is remain zero as no rotation including in the path. It is clear from the figure that when legs in left-hand side in contact, the robot body moves toward right-hand side and vice versa.

4.6.2 Tracking a Straight Sideway Line

In this experiment, the robot will use a sideway walking (crablike walking). In order to achieve this walking mode, the path direction is set up along y-axis and the longitudinal axis of the robot is kept along x-axis. Figure 4.10 shows the crablike walking.

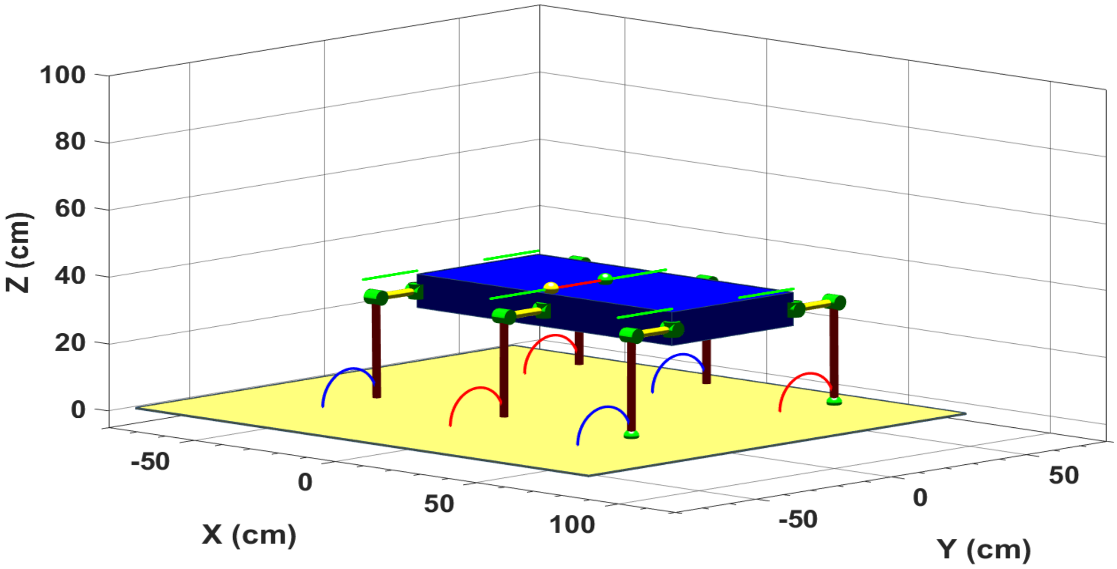
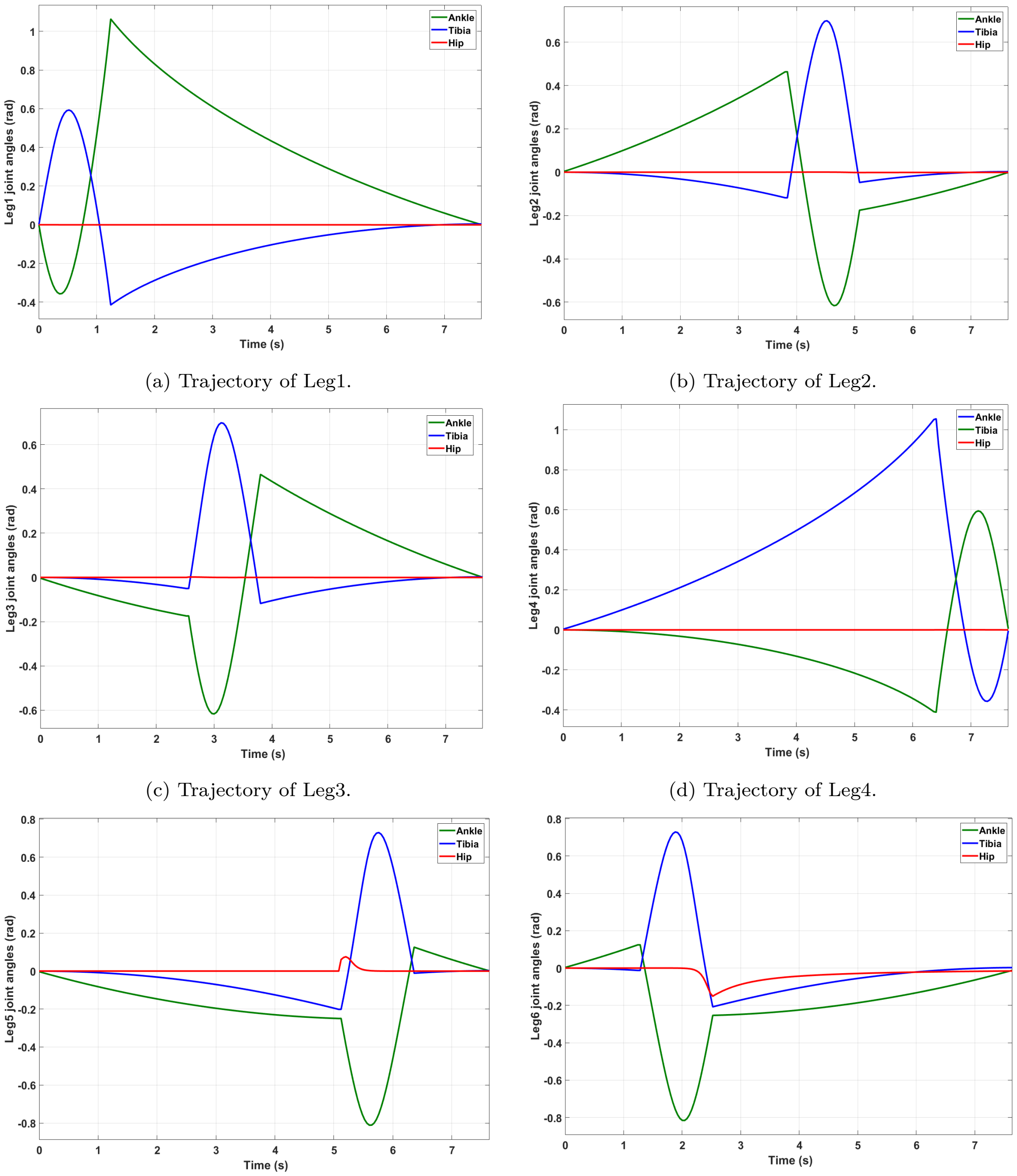


Figure 4.10: Crablike walking.

As the walking path contain no rotation, the trajectories of hip joints of all legs are still constant. In case of the motion path perpendicular on the longitudinal axis of the robot, the values of hip joints equal to zero (no changing), or any other values in oblique direction (remain constant). Both tibia and ankle joints play a significant role in this style of motion. Figure 4.11 shows the trajectory of hip, tibia, and ankle joint of leg one in one cycle. Although, the step size depends on joint ankle, the range of this joint is still within the limit. The tibia joint take part to keep ankle joint inside the bound.



(a) Trajectory of Leg1.

(b) Trajectory of Leg2.

(c) Trajectory of Leg3.

(d) Trajectory of Leg4.

(e) Trajectory of Leg5.

(f) Trajectory of Leg6.

Figure 4.11: Trajectory of hip, tibia, and ankle joints of legs, case of crablike walking. Both tibia and ankle joints slow down gradually to reduce the impact of contact. The hip joints of leg5 and leg6 have changed slightly to compensate for body altitude. All legs joints have returned to the initial values, which means the robot configuration has been retrieved.

As mentioned in the previous experiments, the foot tip should stay steady at support time to avoid slippage. Figure 4.12 demonstrates the position of leg one foot-tip in both x and y direction. Although, the body is propelled at stance time by four legs, the foot of leg one stays fixed at same position.

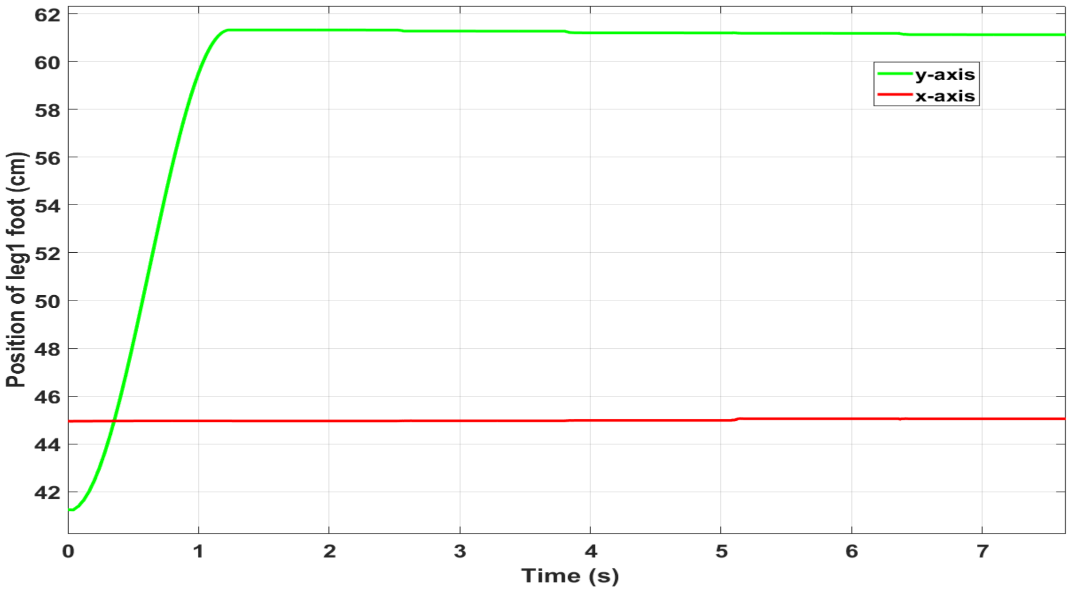


Figure 4.12: Position of Foot Tip along x and y axis. The position 45, 61 cm is conserved during the support phase (time 1.2-7.6 s).

Figure 4.13 illustrates the velocity of the robot in y direction. Although, there are spikes in the velocity profile, the overall speed is still continuous.

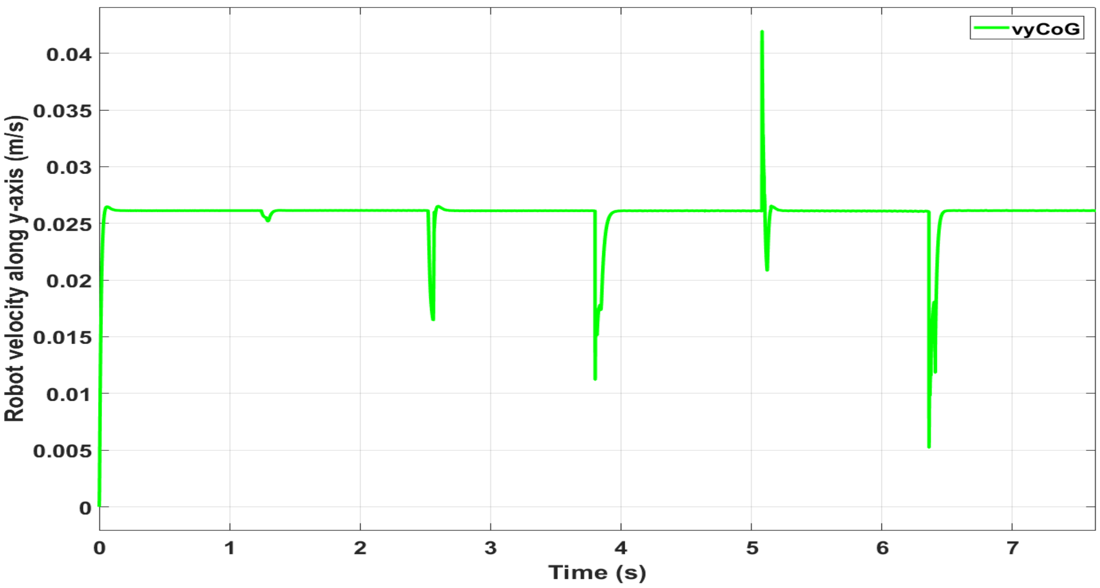


Figure 4.13: Velocity of the robot foot tip along y-axis. The effect of spikes after transition time is very little because it does not become zero over one cycle except at the last step.

4.6.3 Turning to Left

Legs are the source of the motion in legged robots. Controlling the walking direction is achieved by the response of each leg to the direction angle. Since the orientation of the robot's body is defined by the path at CoG, therefore, the contribution of each leg will be different in case of turning. In this experiment, turning to the left direction by 10 degrees will be considered to explain that the generated joints trajectories are varied. All legs should synchronise to keep smooth tracking of the body path. Figure 4.14 shows a top view of the simulated robot, and the robot travels for 20 cm in 7.6 s time. In order to move this distance with 10° of rotation angle around z-axis, several stride lengths for each leg are created. Moreover, the initial configuration of each leg should be recovered; as a result more constraints are imposed to generate the overall motion. Figure 4.15 explains the initial and final position after rotating by 10°, and the path of each leg to perform turning to left motion is illustrated in Figure 4.16.

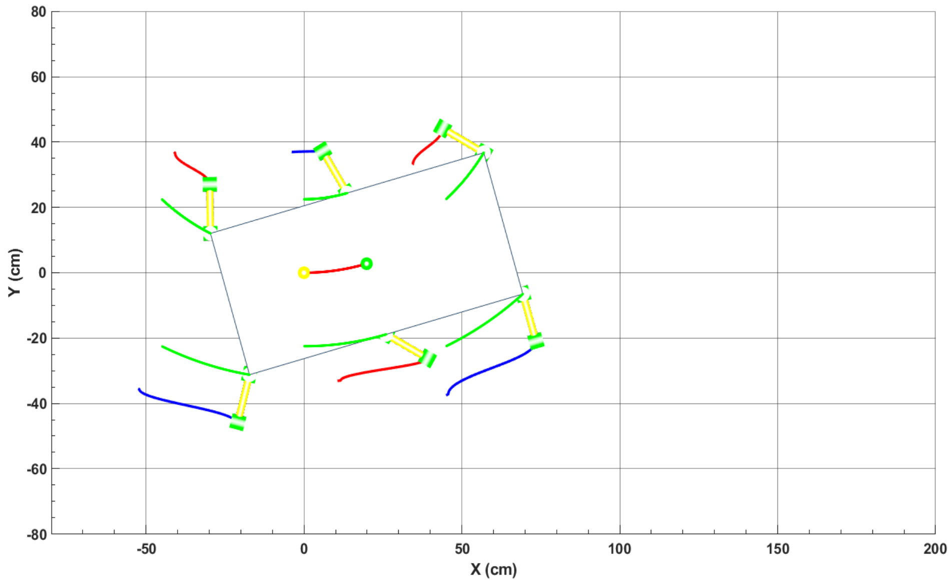


Figure 4.14: Top view of the robot. The robot performs walking with rotation angle equal to 10°. Since the rotation to the left direction, the base of right legs will travel longer distance than left legs.

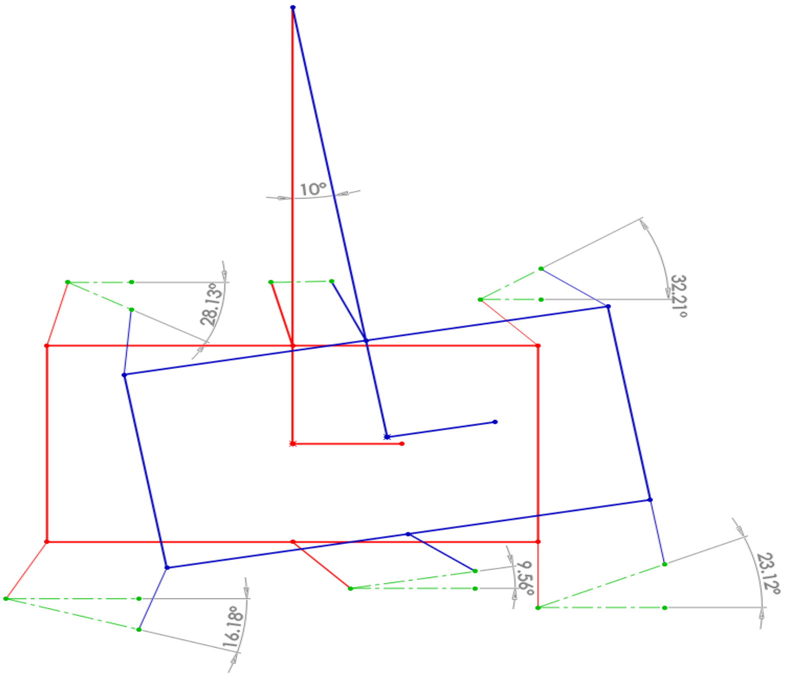
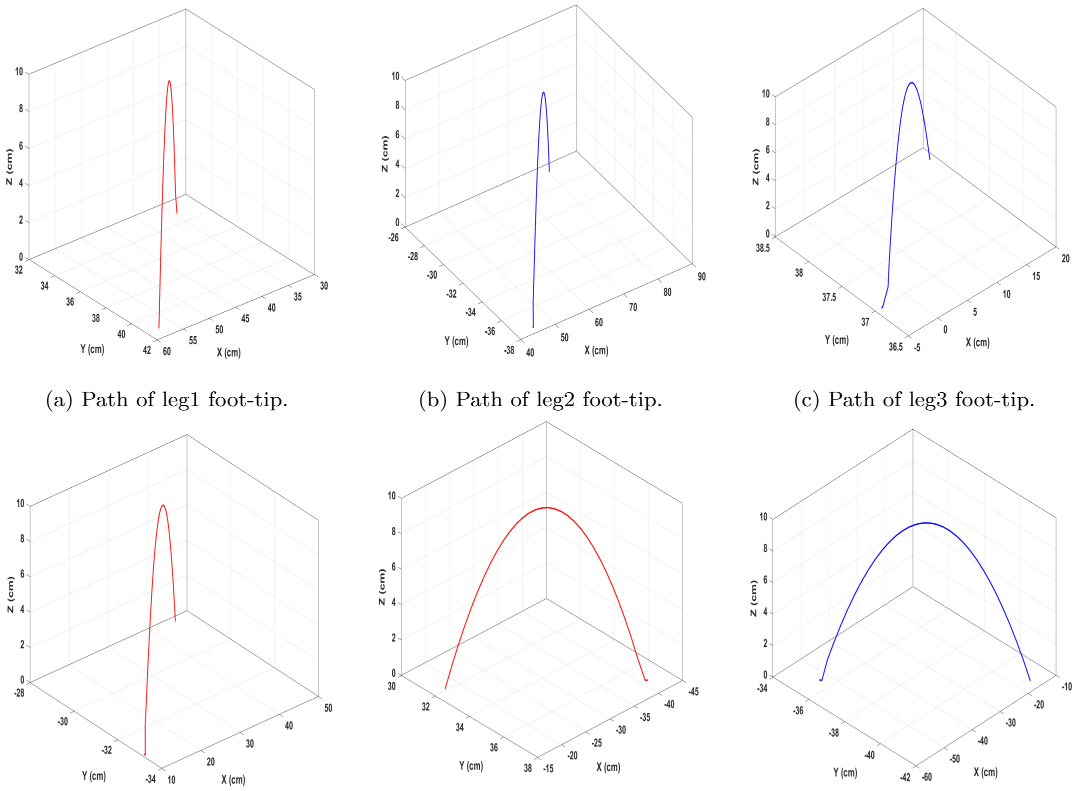


Figure 4.15: Generating trajectory with 10° of rotation around z-axis.



(a) Path of leg1 foot-tip.

(b) Path of leg2 foot-tip.

(c) Path of leg3 foot-tip.

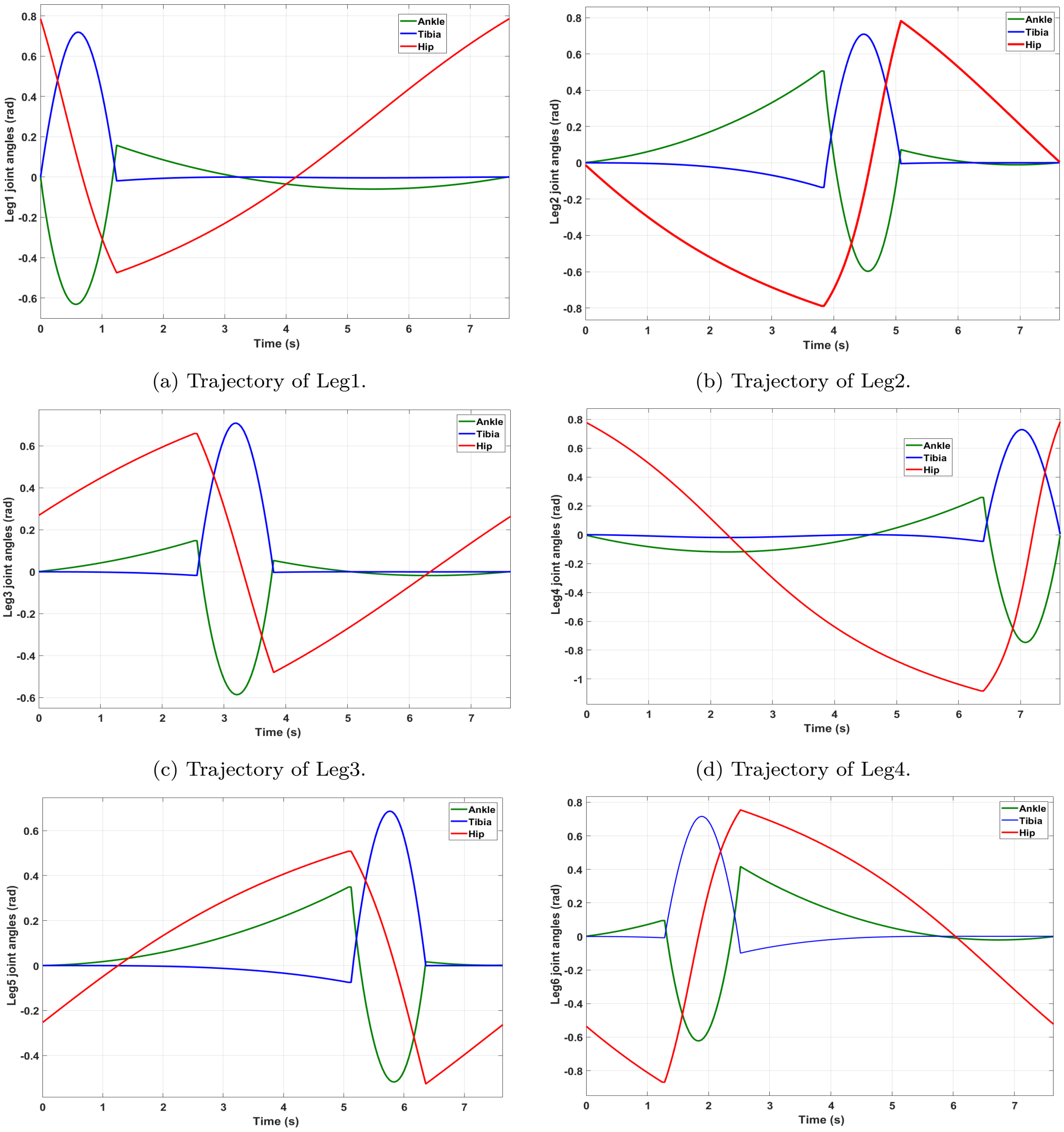
(d) Path of leg4 foot-tip.

(e) Path of leg5 foot-tip.

(f) Path of leg6 foot-tip.

Figure 4.16: Path of all legs, case of turning to left.

In this walking style, each leg has its own trajectory, and the step size of the right hand side legs is bigger than the step size of the left hand side legs.ntribution of joint trajectories for all joints specially in legs that are placed in opposite side to the centre of the rotation angle is bigger than the lags that are placed in the same side of the centre of the rotation angle. Figure 4.17 presents the joints trajectories of all legs. Since the robot rotate towards left side, the hips joints angles of legs (2, 4, and 6) were bigger than the legs in the other side. Same situations can be noticed with ankle joints angles in the right side which were bigger than the left side legs.



(a) Trajectory of Leg1.

(b) Trajectory of Leg2.

(c) Trajectory of Leg3.

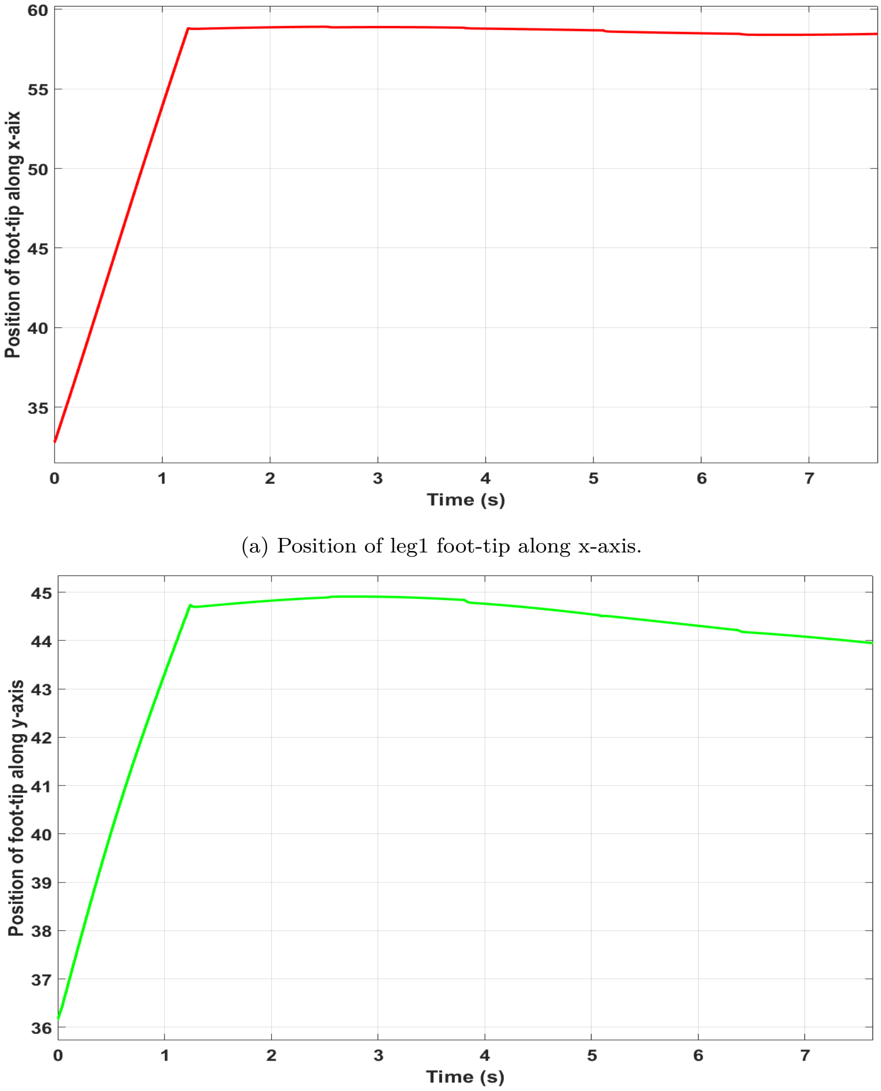
(d) Trajectory of Leg4.

(e) Trajectory of Leg5.

(f) Trajectory of Leg6.

Figure 4.17: Trajectories of joints angles of all legs, case of turning to left.

In case of turning, the propelling of other legs on leg 1 was in two directions (x and y); hence, the position of foot tip has been changed, Figure 4.18 explains the variation of foot tip of leg 1 along x, y direction. In spite of these changes in the leg's foot position (1.9 cm in x-axis and 0.6 cm in y-axis), the position of the robot CoG was accurately tracked.

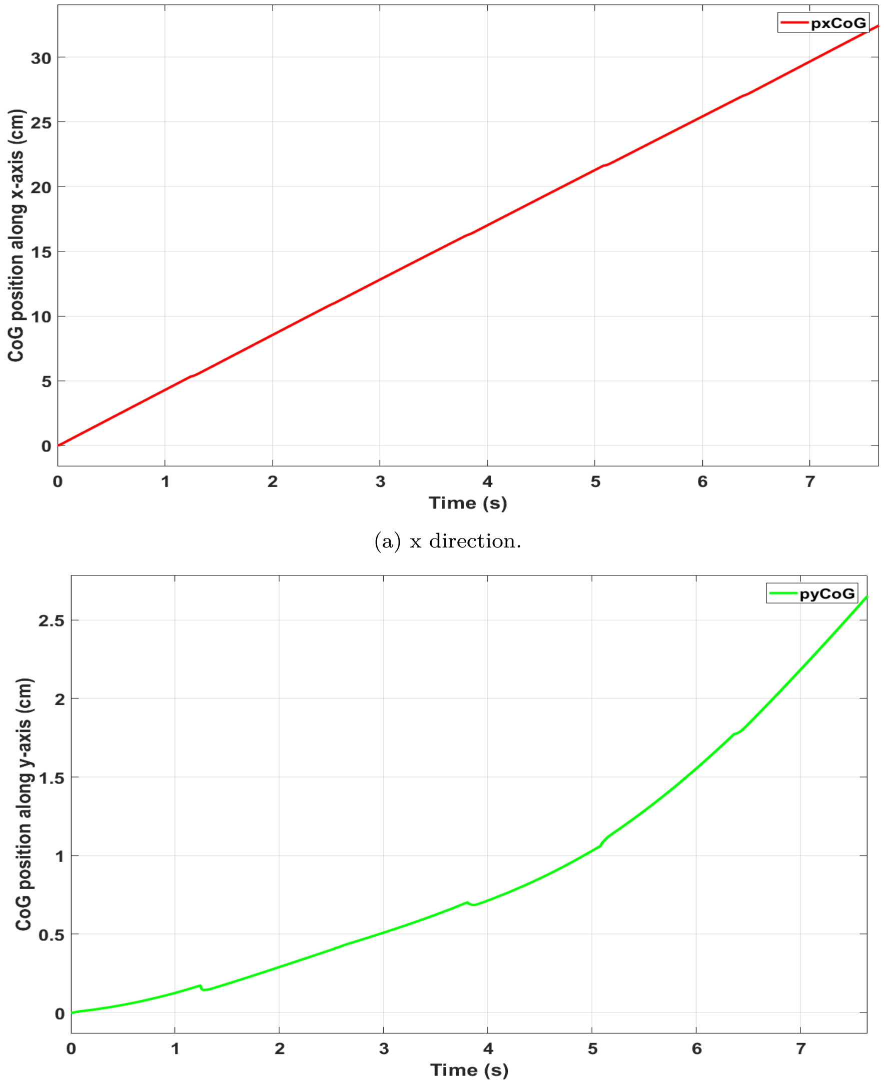


(a) Position of leg1 foot-tip along x-axis.

(b) Position of leg1 foot-tip along y-axis.

Figure 4.18: The position of leg1 foot-tip in x, y directions.

Figure 4.19 shows the position of CoG in turning to left mode, the robot travelled 20 cm in x direction and 3 cm in y direction as a result of 10° turning around z-axis.



(a) x direction.

(b) y direction.

Figure 4.19: Position of CoG in x and y direction.

Same result has been achieved in case of walking in a straight line along x-axis by this mode of motion. The only difference is that the velocity in y direction is raised from 0 - 0.09 m/s. The swings of the body while walking has effected on the velocity in this direction, as shown in Figure 4.20.

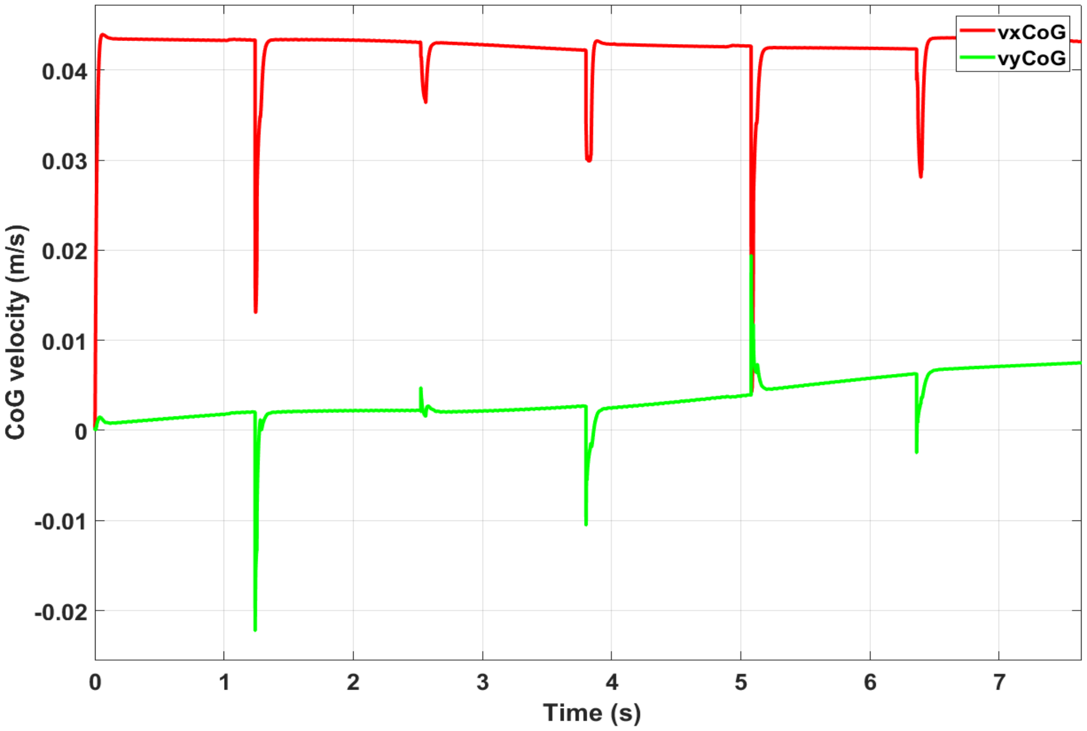


Figure 4.20: Velocity of the body in x and y direction.

4.7 Summary

While walking the six legged robot has to accomplish multiple tasks and satisfy some constraints at the same time. These tasks are represented as a set of linear equality and inequality functions, and a decoupling should be imposed between them according to their importance.

In this chapter, a QP algorithm has been used to handle multiple tasks in hexapod robot. The main task is to achieve a continuous walking with duty factor of 5/6, and to get back the initial configuration of all joints of the robot. In addition, the system has to satisfy more constraints such as joint limit, joint velocity limit, and zero velocity at the contact points. A control point with prescribed reference trajectory has been chosen at the CoG of the robot to determine its direction and orientation.

In order to evaluate the efficiency of the controller, three experiments were performed on the robot: tracking a straight-line along x-axis, tracking a sideways line along y-axis, and turning to the left with constant speed. The results of the experiments demonstrated the efficiency of using QP algorithm to handle both types of constraints (equality and inequality) in any level of priority.

Chapter 5

Sensor-Head Trajectory Tracking

5.1 Introduction

Tracing desired paths and avoiding obstacles by a fixed-base robot arm are common assignments [152], many controllers with different approaches are proposed to achieve these tasks [153, 154]. Typically, the main tasks that assigned to a traditional manipulator arm are achieved by its end-effector [155]. These tasks, for instance, a trajectory tracking, require controlling the position and the velocity of the robot end-effector, and any movement associated with the end-effector is related to the robot-base, which is fixed to the ground. In other words, there is no concern about the position and the orientation of the robot's base. Although, there are many applications proposed a manipulator arm carried by a mobile robot [156, 157], the manipulation tasks are achieved during the stopping period, *i.e.*, no motion associated with the robot's base throughout the manipulation task. The function of the robot is only to allocate a new space for the manipulator to work; consequently, the motion task and manipulation are performed separately in different period of time.

Carrying out similar missions by an arm mounted on a mobile robot are more challenging, if these tasks are associated with robot motion, as the robot-base is not stationary. Lacking this property in floating-base robots implies that the position and the orientation of the robot-base should be considered before taking any action at the end-effector [153]. Since the arm is fixed directly to the robot's body, the position of the arm base is equal to the position of the body's CoG plus certain values in x, y, and z directions according to the body dimension. One possible solution is to take the path of the body into account as a constraint to generate the path of the end-effector. However, while walking over uneven terrain, the robot might encounter unexpected conditions, which lead to disturbing the reference path.

Although the robot's legs play a significant role to maintain the location of the arm-base, they cannot totally guarantee to keep the arm base in a certain position and orientation. Therefore, the arm's joints should be engaged in compensating for any encountered uncertainties at its base; therefore, a complementary role should be initiated for the arm to integrate the operations of both motion and manipulation in the suggested six legged robot.

According to the demining stander operations procedures (SOP) [158], the sensitivity of landmines detector depends on scanning speed and the vertical distance between the sensor-head and ground [159]. The average speed for mine detector to pass over ground is 0.2 m/s, and the typical vertical distance that the detector should stay above the surface is around 15 cm [160].

In the field of mine detection, a trade-off between the operation speed and accuracy should be considered [7]. To speed up the process, the motion of both robot body and the manipulator should be accomplished simultaneously, rather than sequentially [4]. The end-effector should move from one side to the other in continuous motion, and the sensor-head must remain in a fixed distance with the ground to get an accurate measurement from the mine detector. In order to preserve the continuity, a new path planning is proposed for the robot's arm. This path is created from straight-line segment to transit from one side to the other side and a semi-circle curve to move forward.

In this chapter, the requirements to integrate tracking the path of the sensor-head and the path of the robot's CoG, which is represented by the arm-base are presented. This is achieved by planning the path of EE in the operational space and formulating the variation of the arm-base position and orientation as constraints should be satisfied while creating the sensor-head trajectory. In order to perform this task, a 6 DoF's manipulator arm with metal detector is designed and modelled.

To give a clear vision about delimitation of the minefield, designing a virtual environment and a short review about the state of the art in the field of exploiting the robots in demining application is presented in Section 5.2. In Section 5.3, the robot's arm kinematic modelling is explained. Planning the trajectory of the arm EE is introduced in Section 5.4. Finally, in order to verify the effectiveness of the controller and algorithm, four experiments are conducted on the robot's arm in some scanning scenarios.

5.2 Demining Robot and Virtual Environment

Landmines detection activities have seen, as other modern application, as replacement of human by machines. While exploiting mechanical tools in industries aims to speed up production or obtain accurate results, the main objective of utilising machines in demining is to preserve the lives of the people involved in such very dangerous tasks. The secondary objective, but also significant, is to accelerate the scanning operation.

In the context of landmines detection, there are sustained efforts to design efficient robots with the ability to navigate through hazardous areas. The attempts to use wheeled robots are restricted by many limitations make them not applicable in minefields; for example, when detecting a landmine, the scanning operation will be suspended if no choice to avoid it.

Due to its effectiveness, legged robots are preferable over wheeled or tracked robots in demining application as they only require a limited number of contact points with the surface, which can be carefully chosen; this ability reduces the probability of triggering landmines [42]. Furthermore, legged robots are inherently omnidirectional, which is a useful characteristic when manoeuvring in small space [5].

In order to delegate a robot to do these hazardous tasks, a very efficient controller should be implemented in the robot to give the ability to manoeuvre when environmental uncertainties are presented (e.g. uneven terrain) [161]. Most robots that are used in demining applications are controlled by the configuration-space methods [7, 162]. These approaches do not give the ability to use the whole DoF of the robot efficiently [136]. Development of agile and safe legged robots requires designing the whole-body motion in the operation-space rather than the configuration-space [57]. The task function approaches [58] are proposed to design the motion in a space dedicated to the task to be performed and mapping the reference path to the robot's joint space.

The robots used in demining applications still accomplish the process sequentially [163, 7]; in other words, they perform the scanning task and then the task of moving the robot forward switching from one to the other. This approach makes the detection operation too slow. Estremera et al. [24] designed an algorithm to achieve a continuous motion based on reducing the stability margin, this approach was tested on SIL-06 hexapod robot. However, the approach was focused on walking and avoiding hazardous points by the robot legs and no integration with a scanning operation by the arm is mentioned.

To speed up the process of the demining, both scanning task and moving the robot forward should be accomplished simultaneously; this will complicate the design of the robot especially when it is required to keep the end-effector moving with a constant velocity as possible [16]. Therefore, instead of designing the end-effector trajectory as an arc [163], a linear path is suggested; hence, two lines in adjacent boxes are blended by a semi-circular path to ensure the continuity of the scanning operation. A coupling between the velocity of the robot base (robot body) and the sensor-head velocity is implemented; this coupling will give the capability to accomplish many tasks simultaneously [56].

The assumed minefield is divided into boxes each with 20 × 160 cm in x, and y direction respectively. Figure 5.1 shows a diagram of a virtual minefield environment; the dimensions of each box in x and y direction are chosen according to the diameter of the mine-detector and the distance that the manipulator can reach respectively.

A new strategy for planning the trajectory of the end-effector is proposed to travel from side to side within the red boxes of the minefield. In this section, the advantages of scanning the area in front of the robot in a straight-line are discussed. First, it is standard practice to divide the minefield into rectangular segments so a straight-line path will ensure the end-effector passes over the whole area of each box. As a result, the entire area will be scanned homogeneously. Figure 5.2 shows a diagram for one box scanned by traditional techniques (an arc trajectory). Further, in contrast with an arc trajectory planning, our trajectory planning will ensure a continuous motion to the sensor-head when travelling through boxes. Having achieved this, the jerk caused by rest-to-rest trajectory will be reduced [46]. Consequently, the effect of the manipulator inertia will be minimized and the overall body balance will be ensured.

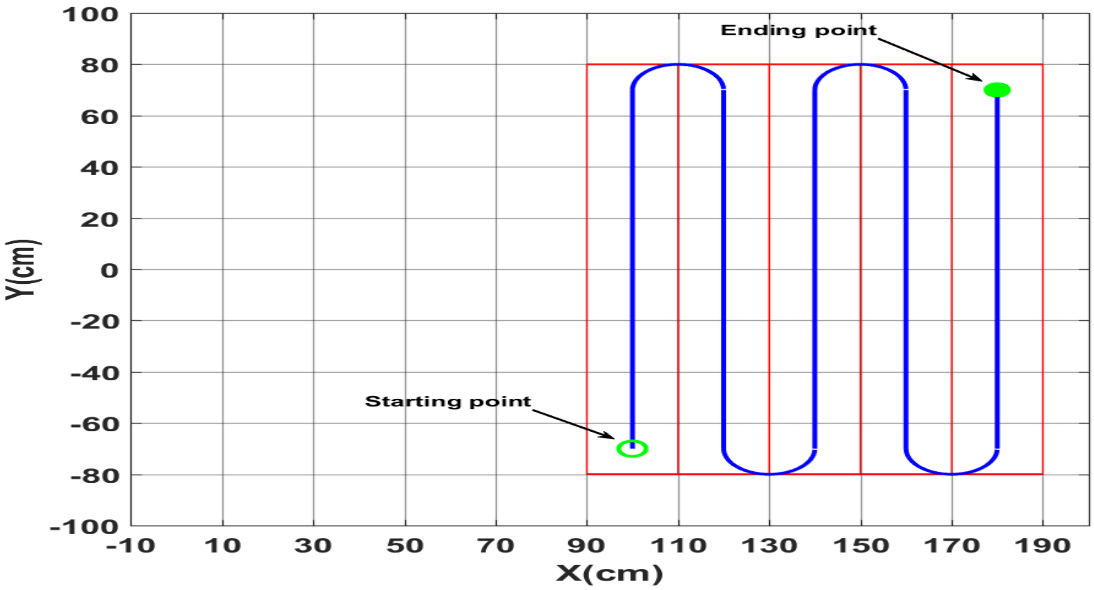


Figure 5.1: Top view for the minefield with five boxes, the blue line represents the path of the end-effector. The green circle and the solid circle indicate the initial position and final position of the whole path respectively. The boxes of the minefield are illustrated by red lines.

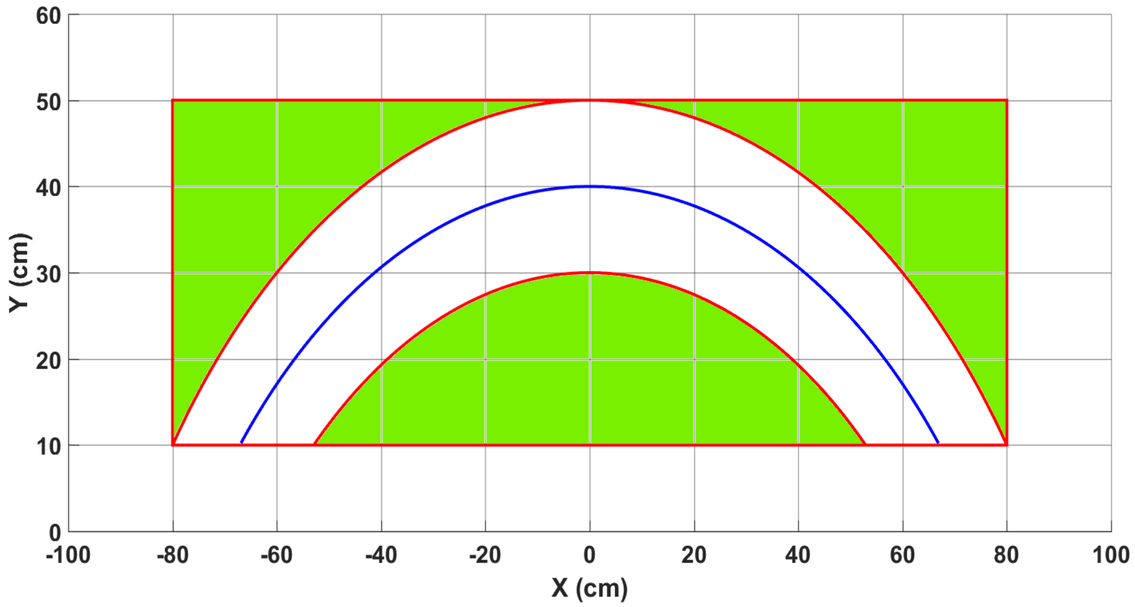


Figure 5.2: Top view of one box scanned by an arc path planning. The blue line represents the sensor-head path, the white region represents the scanned space of the box and the green zone represents the non-scanned area.

5.3 The Kinematic Model of The Robot Arm

A 6 DoF arm is designed to perform the scanning task, this arm is constructed from six revolute joints connected by six links as shown in Figure 5.3. A sensor-head is fixed on the last link, and the arm is connected to the robot's body at joint 1. The transformation matrix between the arm base and the end-effector is formulated using the DH notation. The DH parameters and the joints bound of the manipulator are summarised in Table 5.1.

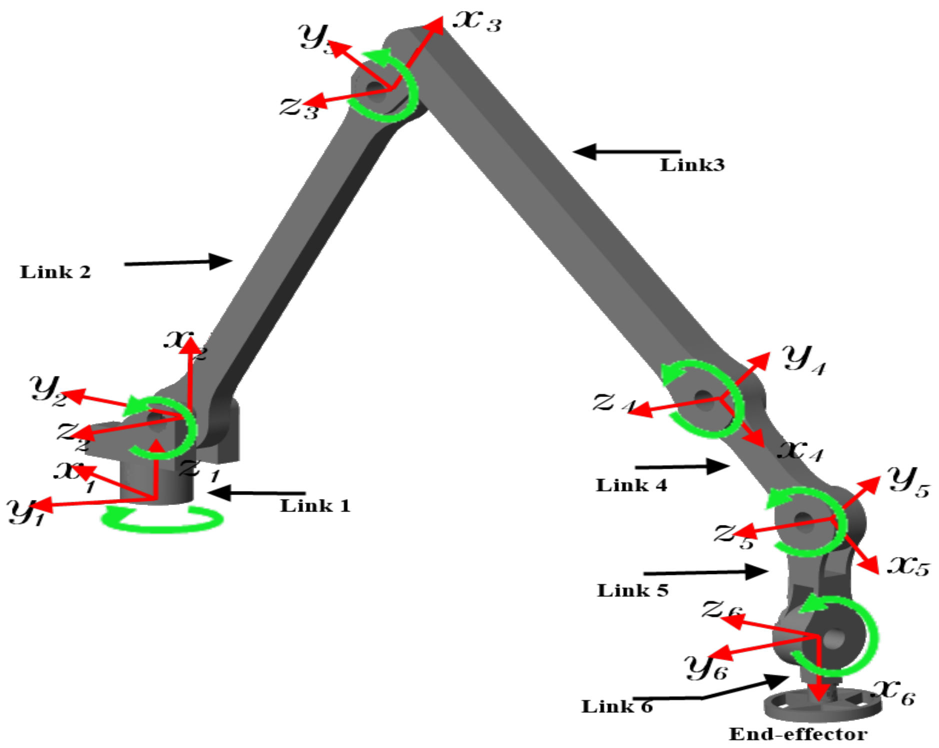


Figure 5.3: Kinematic model of 6 DoF manipulator arm. The frame of each link is fixed at the associated joint.

Table 5.1: DH parameters of the manipulator.

Link	Angle (rad)	d	length (cm)	alpha (rad)	offset	Joint limit
1	q_1	0	0	$\pi/2$	0	(0-90)
2	q_2	0	60	0	0	(0-90)
3	q_3	0	60	0	0	(0-90)
4	q_4	0	30	0	0	(0-90)
5	q_5	0	20	$\pi/2$	0	(0-90)
6	q_6	0	20	0	0	(0-90)

The transformation matrix between link 1 and link 2 is denoted by (T_2^1). In the same manner, (T_3^2), (T_4^3), (T_5^4), (T_6^5), and (T_6^1) are the transformation links for the other links.

$$T_2^1 = \begin{bmatrix} Cq_1 & 0 & Sq_1 & L_1Cq_1 \\ Sq_1 & 0 & -Cq_1 & L_1q_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

$$T_3^2 = \begin{bmatrix} Cq_2 & 0 & Sq_2 & L_1Cq_2 \\ Sq_2 & 0 & -Cq_2 & L_1Cq_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

$$T_4^3 = \begin{bmatrix} Cq_3 & 0 & Sq_3 & L_1Cq_3 \\ Sq_3 & 0 & -Cq_3 & L_1Cq_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

$$T_5^4 = \begin{bmatrix} Cq_4 & 0 & Sq_4 & L_1Cq_4 \\ Sq_4 & 0 & -Cq_4 & L_1Cq_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.4}$$

$$T_6^5 = \begin{bmatrix} Cq_5 & 0 & Sq_5 & L_1Cq_5 \\ Sq_5 & 0 & -Cq_5 & L_1Cq_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

$$T_6^1 = T_2^1\, T_3^2\, T_4^3\, T_5^4\, T_6^5 \tag{5.6}$$

where T_i^{i-1} is the transformation matrix between the link $i-1$ and i.

5.4 End-Effector Trajectory Planning

To satisfy the constraints that imposed by the environment, it is convenient to plan the EE trajectory in the space of the task and then mapping the reference path to the joint space. In order to perform scanning operation, two significant tasks are assigned to the manipulator. First, according the virtual minefield 5.1, the EE moves from one side to the other in $y - axis$ direction, this is achieved by tracking a straight-line path with 5 cm (this height can be adjusted as required) in height with respect to the ground along the vertical direction. In case of an obstacle inside the path, the via-points that constitute the path will be updated according to the dimensions (height and width) of the obstacle. The second task is moving from the current box to the next one using a semi-circular path, which is used to blend two linear paths in adjacent boxes. Although, the geometric nature of the two paths seem to be different, both paths are formed from multiple points, which represent the geometry nature of the generated path. These points are blended by a fifth order spline to get sooth trajectory. A trajectory of the manipulator with an obstacle inside is shown in Figure 5.4.

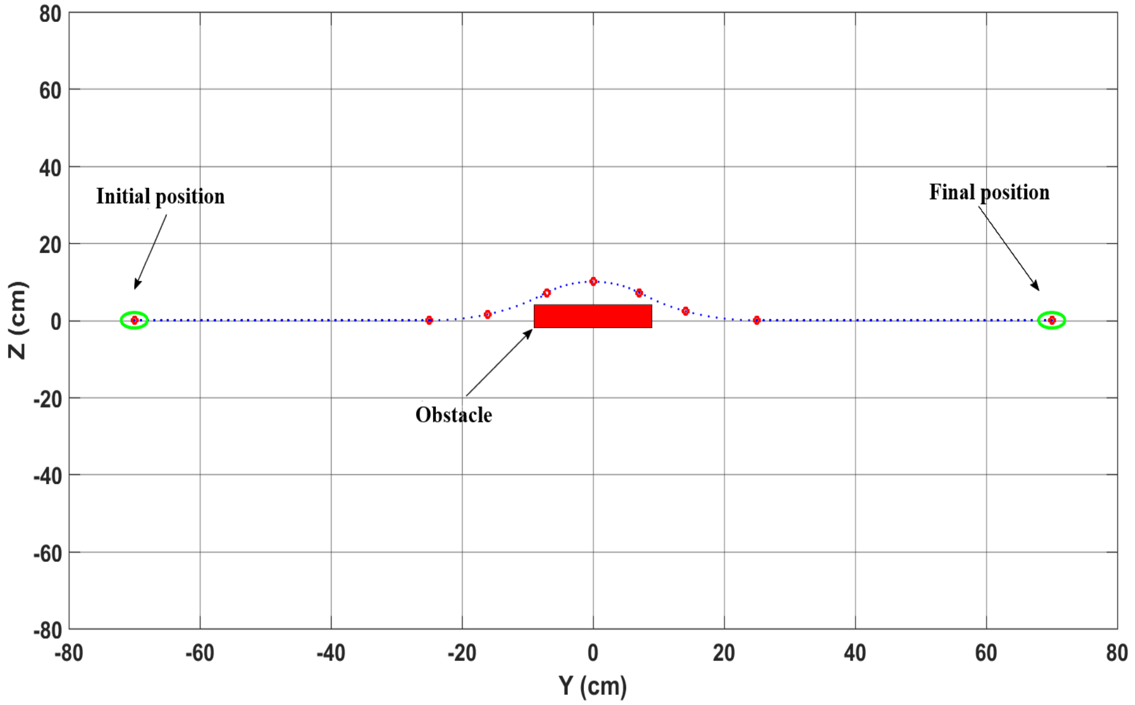


Figure 5.4: Example of a path of the end-effector with an obstacle inside it. The green circles represent the initial position and final position of the end-effector. The seven red circles represent points depending on the dimensions of the obstacle. The path is interpolated by a linear and parabolic polynomial spline.

$$T = \begin{bmatrix} R_o & p_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{5.7}$$

where $R_o \in \Re^{3\times3}$ is the rotation matrix of a certain point with respect to arm-base and p is a translational vector, which describe the path point.

Each point is described by three translational parameters along $x, y,$ and z axes and three orientation angles (roll, pitch, and yaw). The coordinate transformation between a certain point and the arm-base can be represented by (4x4) transformation matrix, Equation (5.7).

The path is initiated by defining a starting point that the EE will begin from. While the values along y direction change, according to the number of the intermediate points along the path, the values of the path along x and z will stay constant. The orientation will stay fixed along the linear segments. In order to facilitate smooth transition at the end of linear path to the next box, both the direction and the orientation will be changed at the semicircle segments. The rotation of each frame in the parabolic segment is around z-axis. The origin of the semicircle path is defined by the ending point of a linear line in y-axis and half the linear distance between two adjacent boxes in x direction, which is equal to 10 cm. Further, the travelled distance by the EE is 140 cm (in y direction), in addition to 10 cm (in y direction) while crossing to the next box. Therefore, planning the trajectory of the EE will ensure the whole area of a box is completely covered by the detector. In addition to the parametric representation of each frame, the motion rate is calculated according to the length of the prescribed path.

The corresponding joint angles are obtained using inverse kinematic methods. Any potential change in the arm-base position and orientation will be compensated by retrieving back the pose of the EE. In this approach, a trade-off between the computation cost associated with the inversion operation and the upper limit of the sampling rate should be taken into account. In order to determine the height and the width of an obstacle, three cameras are presumably installed in certain points on the robot. Two of them are fixed on both sides of the sensor-head. The third camera is mounted underneath the robot body. Figures 5.5, 5.6 illustrate a view of the lower camera to the arm end-effector and the view of the camera which installed on left side of the sensor-head. The information of these cameras is fused to generate the required information in the trajectory planning. More details about extracting and fusing information from cameras can be found in [155, 164, 122, 165].

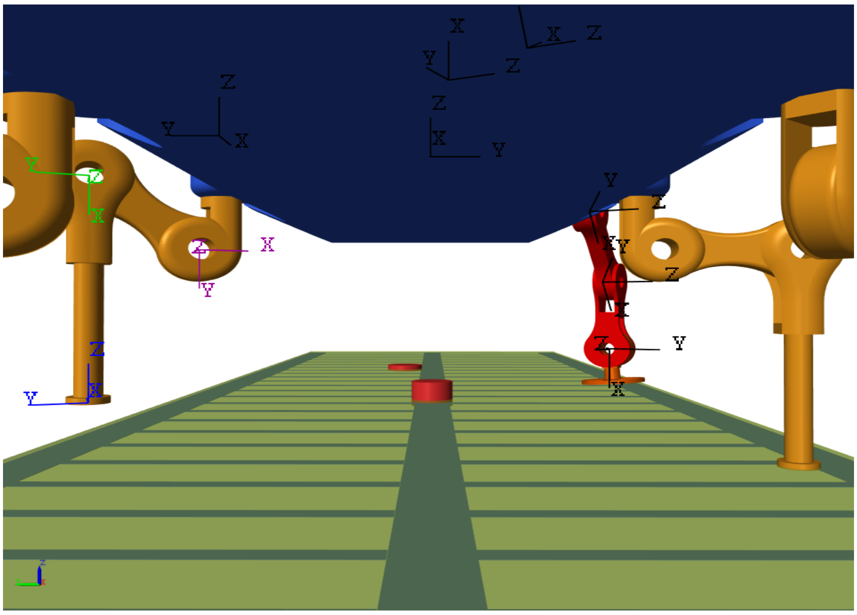


Figure 5.5: View of the camera, which fixed underneath the robot body.

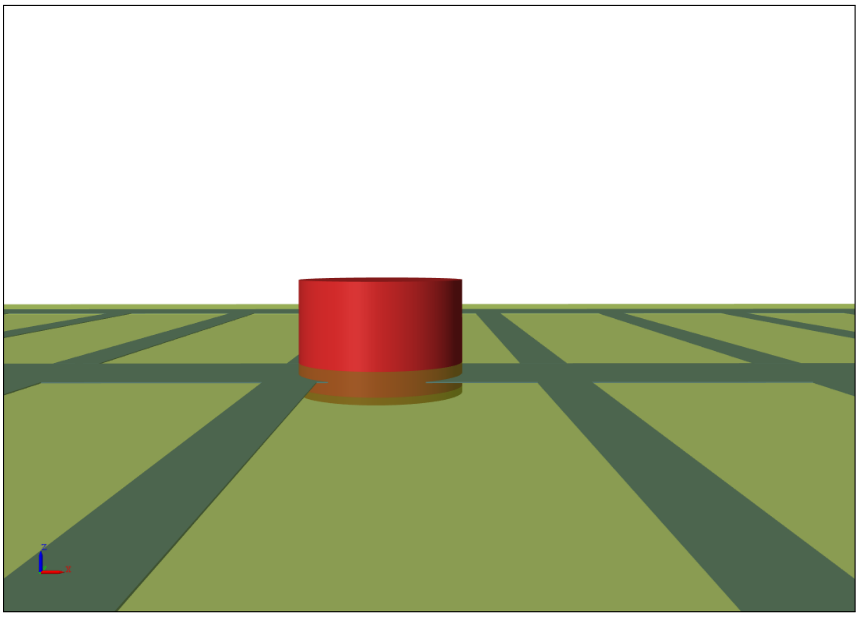


Figure 5.6: View of the camera, which fixed on the left side of the EE.

5.5 Experiments

In order to verify that all the defined tasks are achieved, four experiments have been conducted, and the motion of the robot is considered in each evaluation. In the first experiment, the manipulator follows a straight line path with 5 cm height with respect to the ground at its end-effector. Following a straight line with an obstacle inside is considered in the second experiment. In the last two experiments, the straight-line path of the arm-base is replaced by a parabolic and an arbitrary path to investigate the ability of the EE to track the prescribed straight line path. The results are expressed by the simulated robot-arm inside the virtual minefield, the values of joints angles, the position of the EE (x, y, and z) with respect to the world frame, and the linear velocity of the EE.

5.5.1 Tracking a Path with 5 cm Height from Ground

In this experiment, the arm EE will scan two free obstacle boxes. The initial configuration of the arm is set in a certain way so that the EE starts from the first predefined box, and the initial position of the arm-base is [45, 0, 70] cm in x, y, and z respectively. This position of the arm-base has been chosen in order to mediate the boxes in y-direction, and the initial position of the EE is [100, -75, 5] cm in x, y, and z respectively. Figure 5.7 shows the complete task of scanning two boxes. Snapshots of top view for the sensor-head are presented in Figure 5.8; the figures show five snapshots in a different time periods, the sensor-head position is remain constant with respect to z-axis.

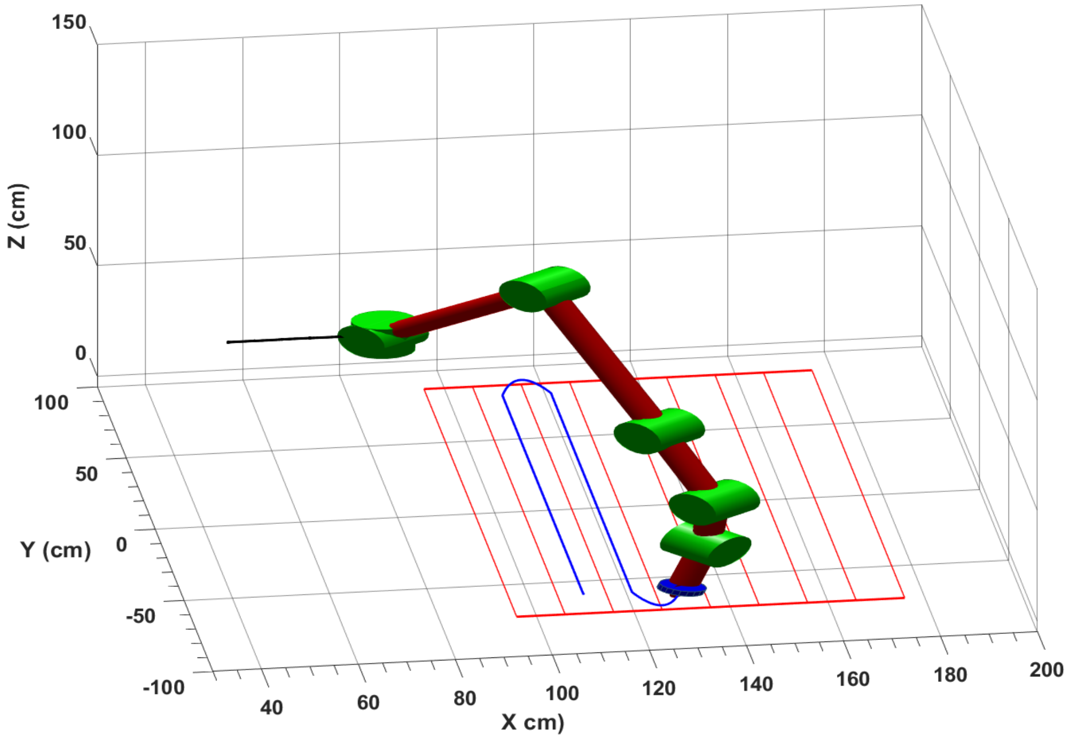
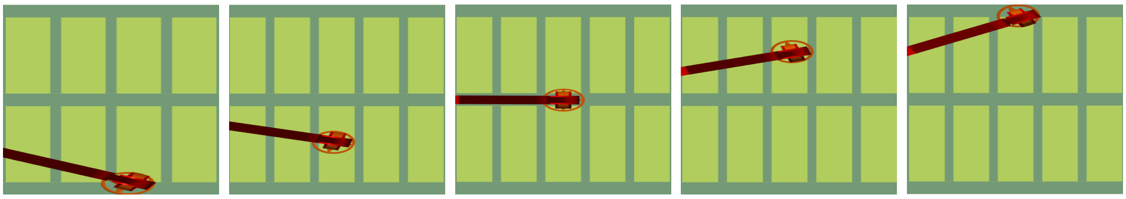


Figure 5.7: Complete task of scanning two boxes. The arm-base follows a straight line and a distance with 5 cm height between the EE and the ground has been maintained.



(a) Time 0.55 s. (b) Time 0.98 s (c) Time 1.38 s (d) Time 1.86 (e) Time 2.21 s

Figure 5.8: Top view for the sensor head. The transmission of the sensor-head in different time steps.

Figure 5.9 illustrates the contribution of each angle of the arm in this task. It is clear from the figure that the arm posture is retrieved at the end of the task.

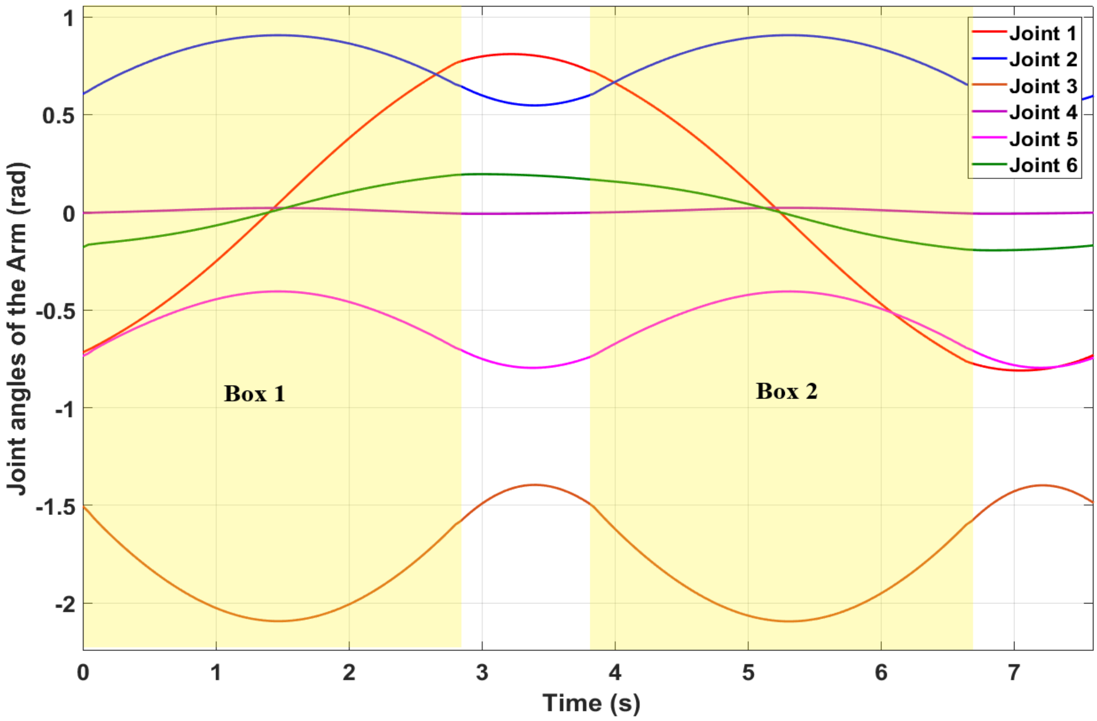


Figure 5.9: Values of the joint angles of the arm for one cycle. The transition to the next box was occurred at time 2.9 s. The highlighted areas represent the time when the arm operate over box 1 and box 2.

Figure 5.10 shows the position of the end-effector in x, y, and z directions. The condition of keeping the EE in a constant level of 5 cm in z direction is maintained, the position of the EE in x, and y has been changed at time 0 - 2.7 s to show the motion of it. When the EE transferred to the next box at time 2.7 - 3.7 s the values in x direction were changed according to the width of the box. The variation of the EE position in y-axis reveals two things:the travelling of the EE in both directions of y-axis and the EE returned to the same initial configuration, which allows the scanning operation to start in a continuous manner.

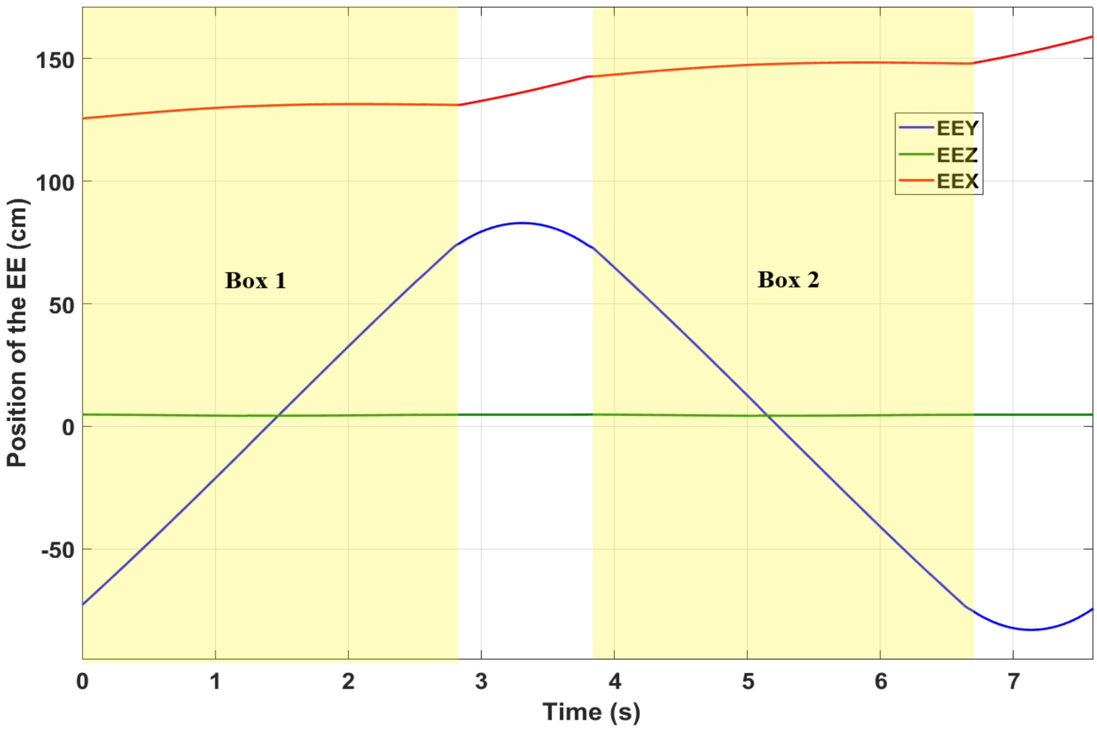


Figure 5.10: Position of the EE in x, y, and z directions. The height of the EE has been kept at 5 cm. The position along x-axis has changed only when the EE was transferred to the next box. The changing in y direction included travelling from -75 cm to 83 cm while scanning the first box and from 83 cm to -75 when scanning the other.

As shown in Figure 5.11, the velocity of the EE in y direction rose from 0 to 0.5 m/s in 0.01 s, this represents the initial velocity. The velocity continued to rise until reaching 0.55 m/s, which represents the maximum speed before changing its direction, when the EE starts to enter to the next box. Despite the change of direction, the continuity of the EE speed was maintained.

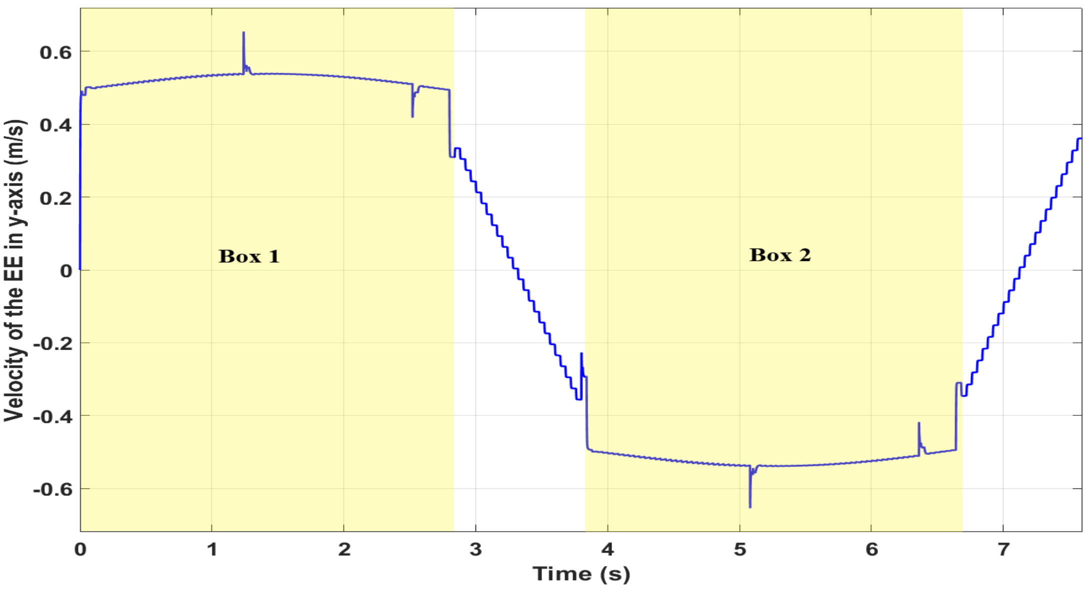


Figure 5.11: Velocity of the EE in y direction.

5.5.2 Tracking a Path with 5 cm Height and an Obstacle Inside

In this experiment, the robot motion is tested with a case of an obstacle inside the first box. The EE path is generated according the dimensions of the obstacle which are captured by the cameras. Three segments constitute the path in the first box. The first segment begins from the EE initial point to a point 5 cm away from the edge of the object. The second segment is a semicircle spline above the object with a diameter 11 cm (5 cm above the upper face of the object). The third segment starting from a point with 5 cm in y direction away from the second segment to the final position. These segments are blended by cubic polynomial to get a smooth Cartisian path. This experiment implies that there are more constraints to satisfy by the controller. Figure 5.12 shows the robot arm scanning two boxes, the first box has an obstacle (landmine with a dimensions of 10 cm, 10 cm, 6 cm in x, y, z respectively). The object is located in 90 cm, 0 cm, 0 cm in x, y, and z directions.

The dimension of the landmine and its location has been chosen for simplicity to evaluate the experiment. Nevertheless, the trajectory generation algorithm has the ability to generate a trajectory for the EE with any dimension and location of an object very quickly.

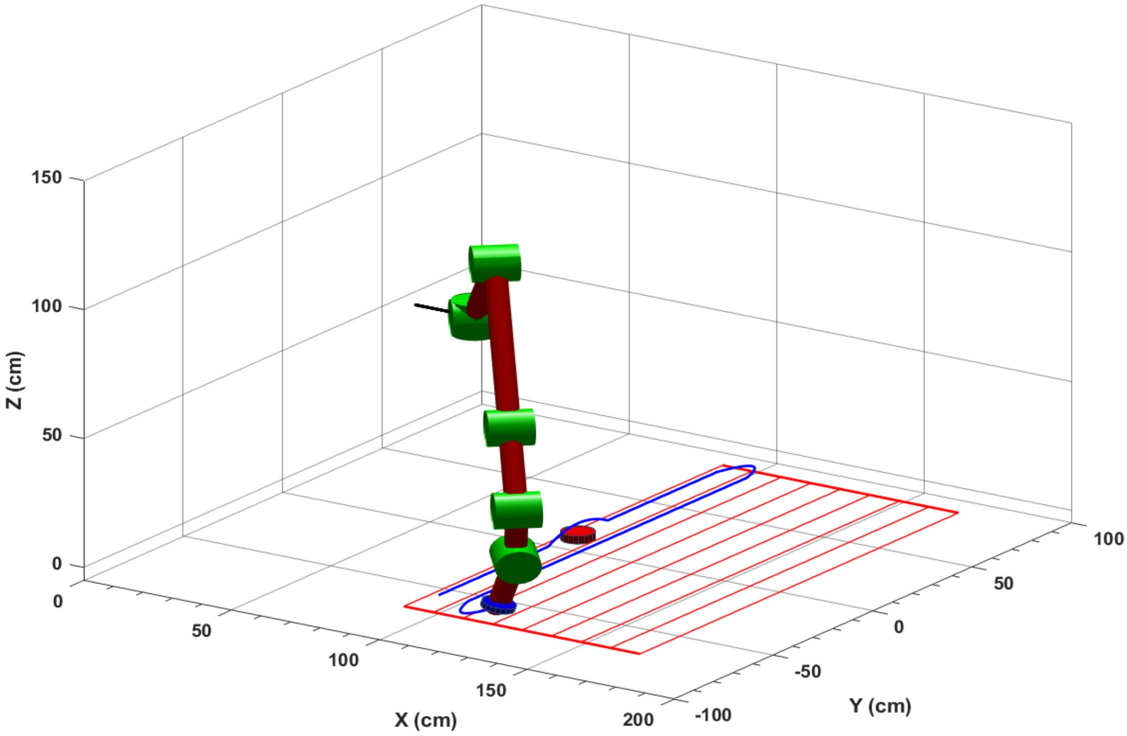
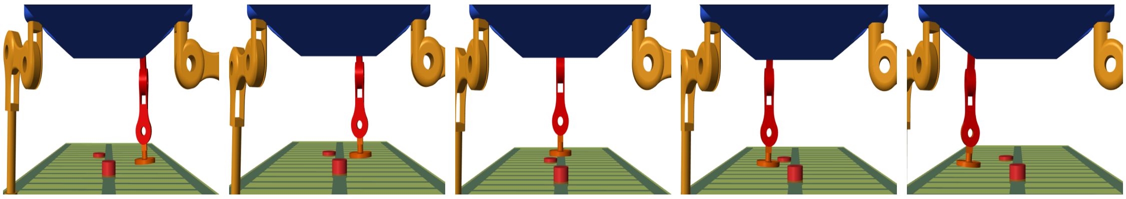


Figure 5.12: Scanning two boxes with a cylindrical shape obstacle in the first box. While the base of the arm follows a straight line, the EE track a path with 5cm in height above ground and an obstacle inside.

Figure 5.13 demonstrates five snapshots in different time steps for the sensor-head before and after avoiding the landmine.



(a) Time 1.2 s. (b) Time 1.3 s (c) Time 1.6 s (d) Time1.8 (e) Time 2.1 s

Figure 5.13: Snapshots taken by the lower camera. The transmission of the sensor-head in different time steps.

Figure 5.14 demonstrates that the initial configuration of the arm is returned back. Further, the contribution of all joints at time 1.1 - 2.1 s is clear in the moment of avoiding the object, which highlighted by light purple colour).

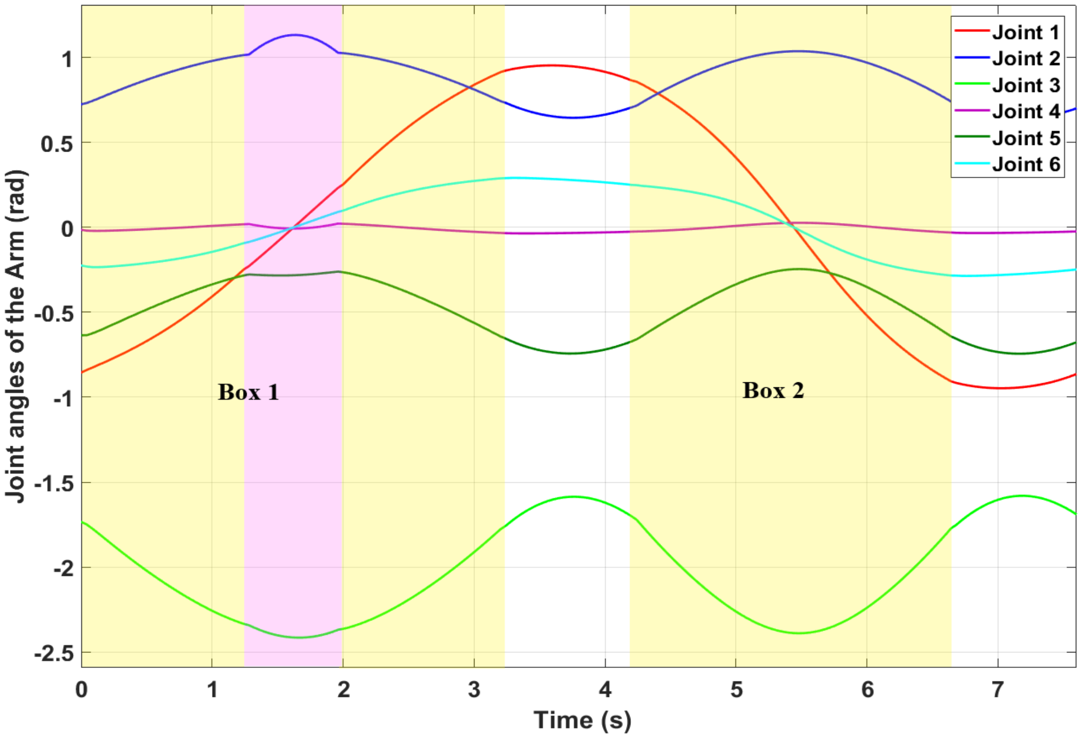


Figure 5.14: Values of the joints angles of the arm. The presence of the object inside box 1 is highlighted by light purple colour to indicate the roll of the joints in avoiding task. The constraint of returning the initial configuration is satisfied.

The position profile of the EE is illustrated in Figure 5.15. The height of the EE is maintained at 5 cm over ground except at time 1.2 - 2.1 s where the landmine is located. The path of the EE along x direction was not affected by the presence of the obstacle.

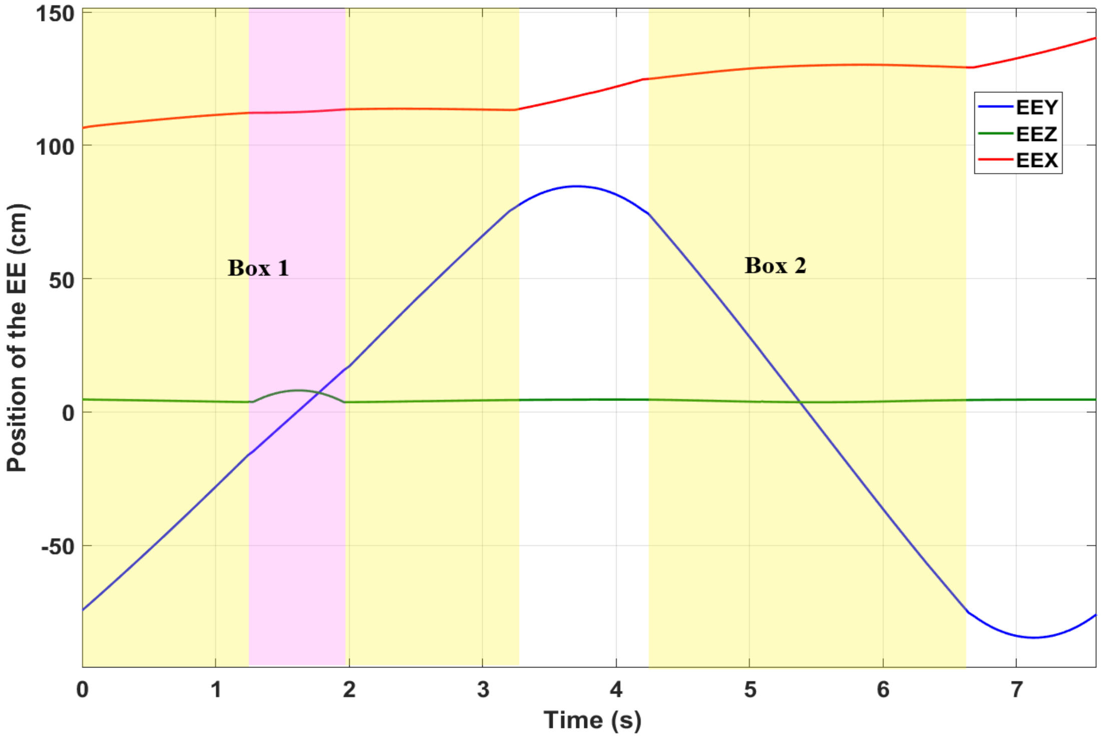


Figure 5.15: The position of the EE in $x, y,$ and z direction. The EE has lifted over the specified level by 5 cm height, which is enough to avoid the obstacle. The positions in x and y direction stayed unchanged as the previous experiment.

The effect of presence of the obstacle on the EE velocity is obvious as shown in Figure 5.16. Although, the velocity dropped down by 0.05 m/s due to the gravity, the average motion rate of the EE remained acceptable.

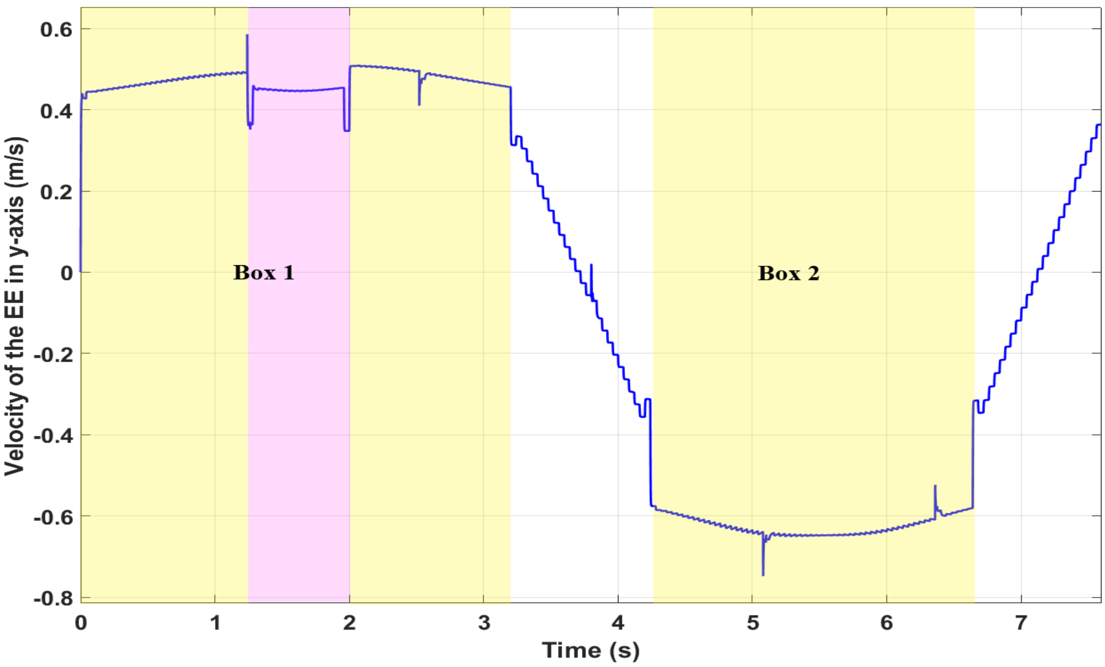
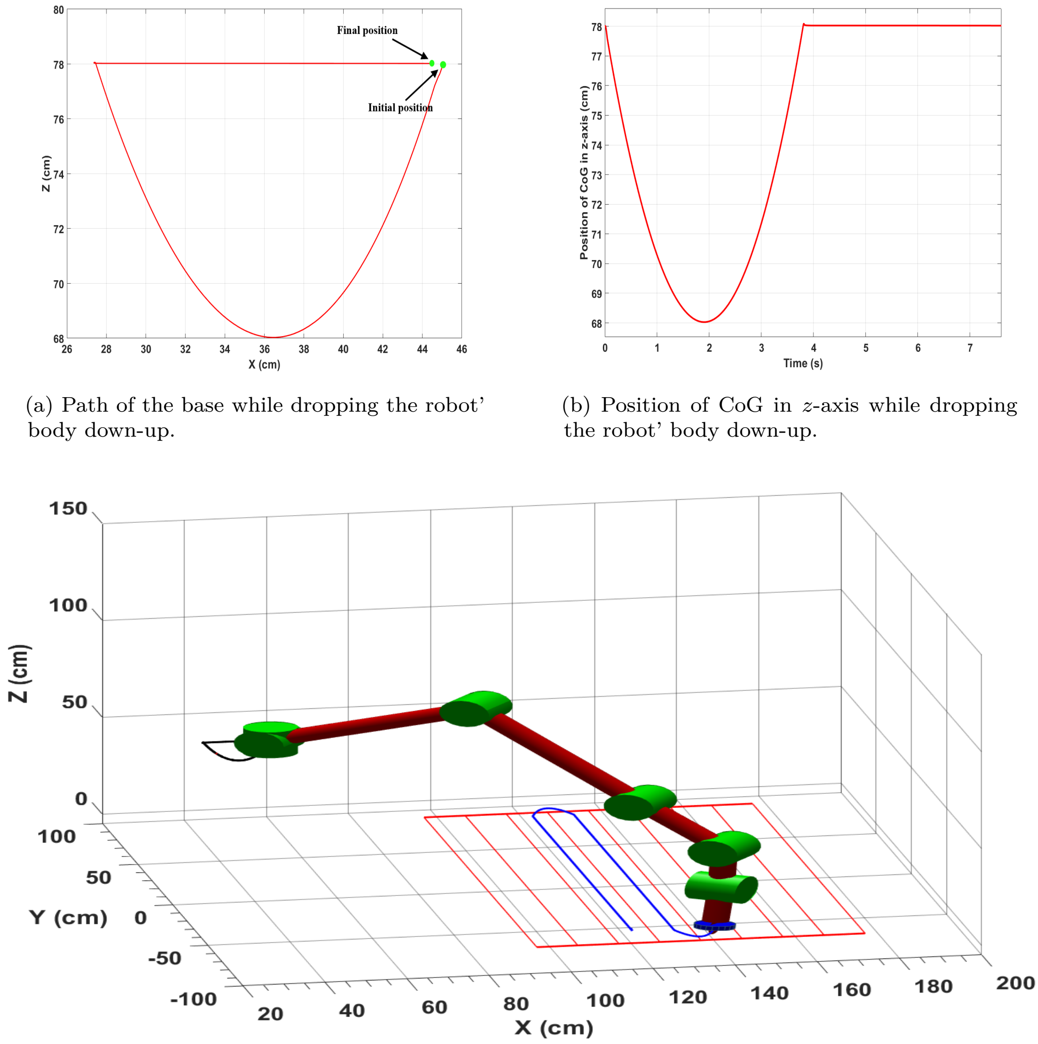


Figure 5.16: Effect of presence of the obstacle on the velocity of the EE. The velocity has dropped down by 0.05 m/s due to gravity.

5.5.3 Gradually Drop in the Arm Base in z-axis Followed by Moving Forward in x-axis

In this experiment, the response of the EE to track a straight-line path while gradually dropping the arm-base down in z direction is investigated. The path of the arm-base is generated in such a way the body of the robot is dropped gradually in z-axis and moved forward in x-axis. In order to get the robot's body down, all legs perform transfer and stance phase at the same time. Technically, if all legs lifted simultaneously, the legs will stay in contact with ground and the body will drop down. Pushing the body forward is achieved by all legs during stance phase. Figure 5.17a illustrates the entire path of the arm base. The path is started from initial position at 45, 0, 78 cm along $x, y,$ and z-axis respectively. The dropping distance in z-axis is 10 cm and the range in x-axis is around 18 cm. The second part of the path is in x-axis only which is started from around 27 - 44.4 cm. The initial and final point is indicated by two green balls. The transition of the CoG along z direction is depicted in Figure 5.17b. The complete scanning for two boxes by the manipulator while drooping in z direction is shown in Figure 5.17c and top view snapshots for the sensor-head are illustrated in Figure 5.18.

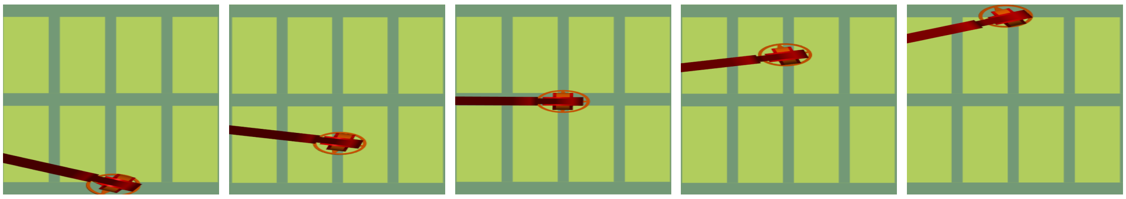


(a) Path of the base while dropping the robot' body down-up.

(b) Position of CoG in z-axis while dropping the robot' body down-up.

(c) Scanning two boxes while dropping the arm base down-up and moving forward.

Figure 5.17: Path of the arm base case of gradually dropping in x, z-axis and moving forward. The initial and the final position of the path are indicated by green balls.



(a) Time 0.55 s. (b) Time 0.98 s (c) Time 1.38 s (d) Time 1.86 (e) Time 2.21 s

Figure 5.18: Top view for the sensor head. The transmission of the sensor-head in different time steps.

As depicted in Figure 5.19 both the joint limit and the initial configuration conditions are perfectly maintained.

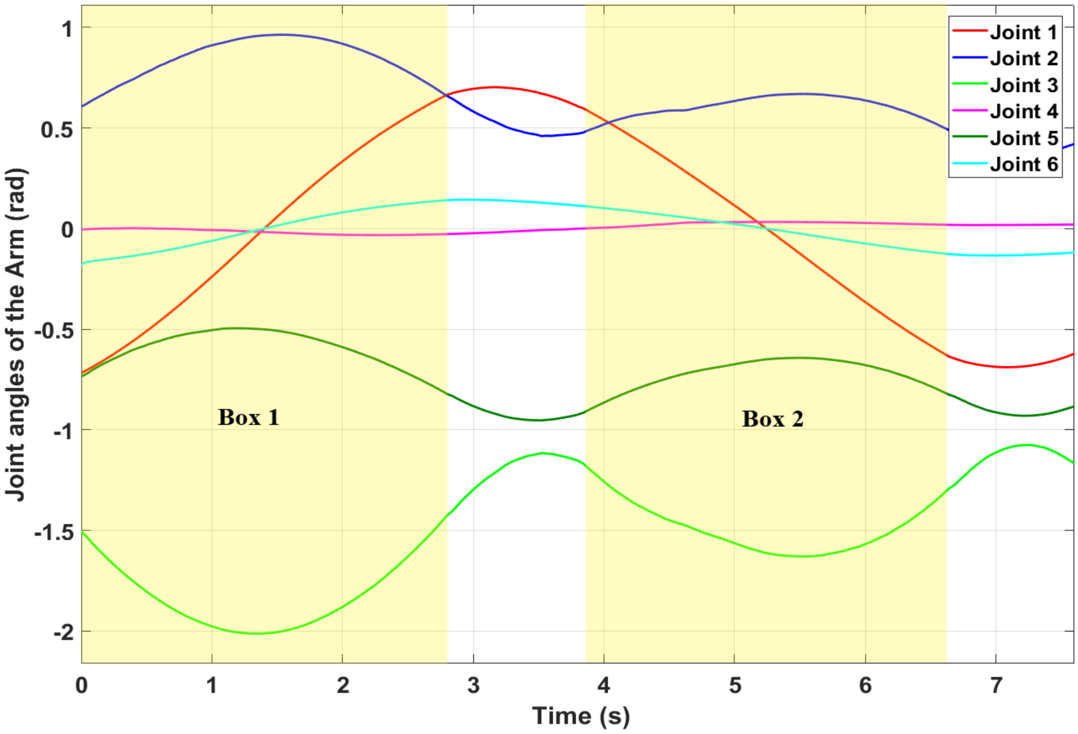


Figure 5.19: Joints angles of the arm case of gradually dropping down-up in z direction and moving forward.

Although, the average speed of dropping the body down is 5 cm/s, the altitude of the EE in z-axis has no affected except for a short time at 0.4 - 1.3 s, as shown in

Figure 5.20. If it is compared with the initial position of the EE along z -axis with respect to ground (5 cm), the response of the controller is good.

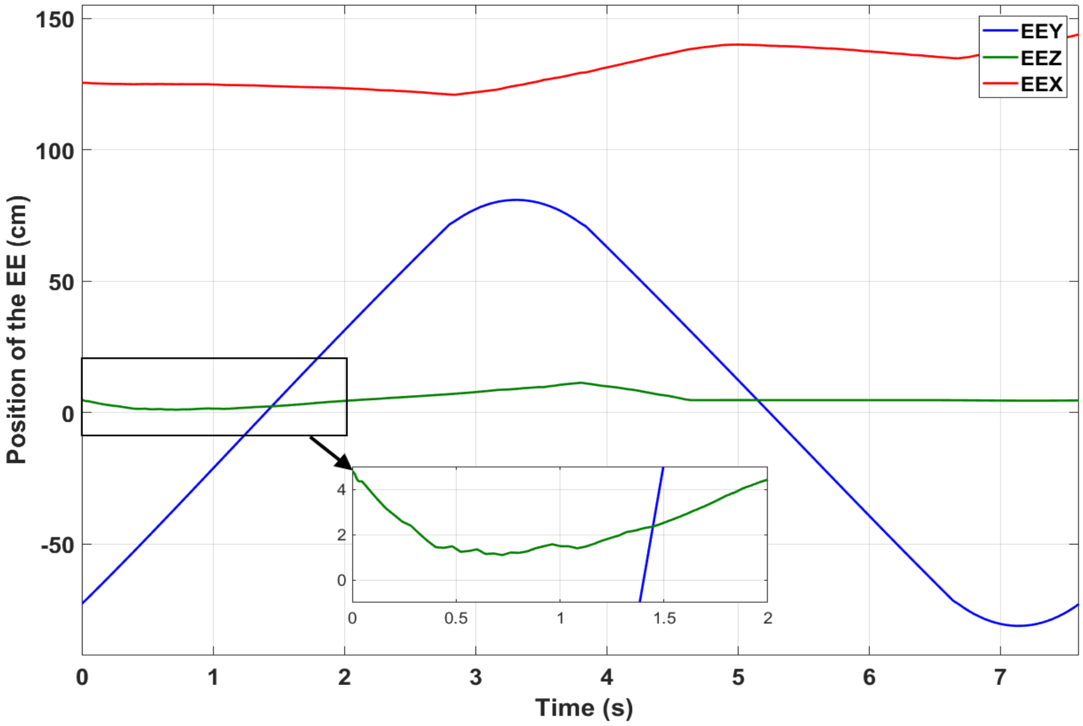


Figure 5.20: EE position in $x, y,$ and z direction case of gradually dropping in x, z -axis and moving forward. The position of the EE in z -axis between time 0 - 2 s is magnified to illustrate the response. The distance between the EE and the ground is still more than 1.5 cm.

Figure 5.21 demonstrates the average speed of the arm base in x and z directions. Due to the body motion was started to backwards, the velocity in x direction remain constant until time 3.7 s when the body was returned back to the initial height. After time 3.7 s, the velocity in z direction settled down to zero, as there was no motion occurred.

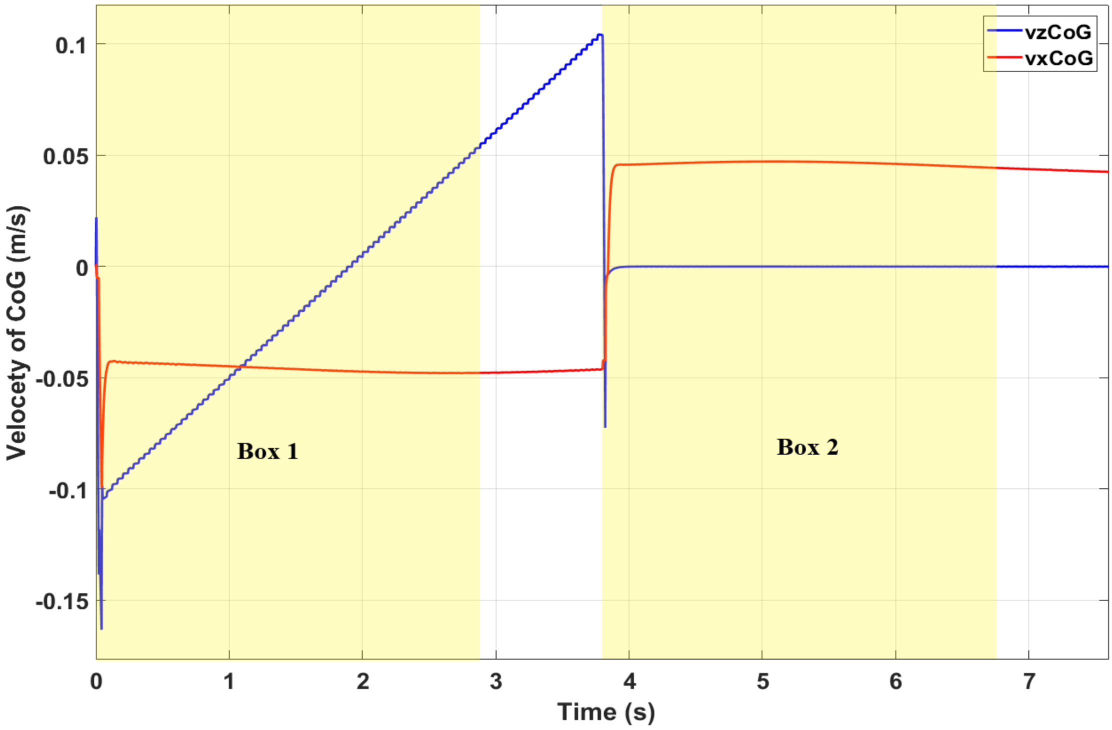


Figure 5.21: Arm base velocity in x and z directions.

The velocity of the EE is illustrated in Figure 5.22. It is clear from the figure that the changing in the arm-base in one direction (z-axis in this case) has no crucial impact on the continuity of the EE motion.

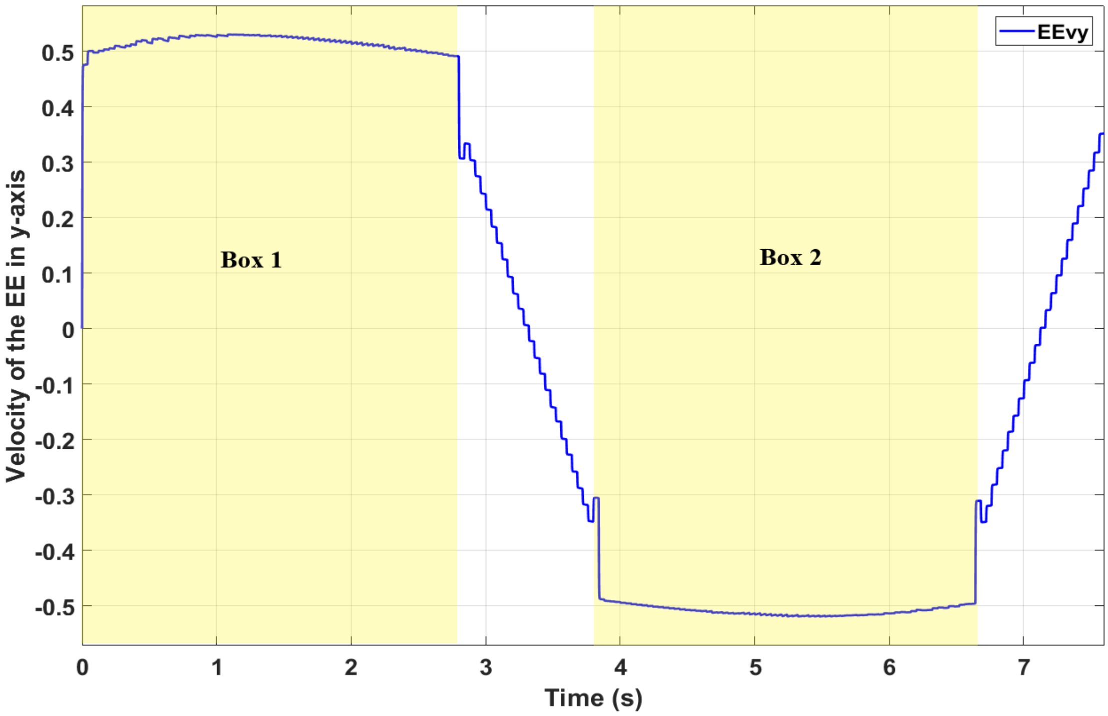


Figure 5.22: Velocity of the EE case of gradually dropping in z-axis and moving forward in x-axis.

5.5.4 Arbitrary Base Path

In general, walking over rough terrain causes changing the altitude of the robot's body randomly. In order to investigate the effectiveness of the controller to cope with arbitrary change in the base position, a path as shown in Figure 5.23 is generated. This path is created by making the legs in the left-hand side (legs 1, 3, 5) and the legs in the right-hand side (legs 2, 4, 6) to move in different motion phase. The range of the arm-base motion is 45 - 43.6 cm in x direction, -3.5 - 2.9 cm in y direction, and 75 - 78 cm in z direction. The goal of this experiment is to test the effect of changing the arm-base in three direction in the same time.

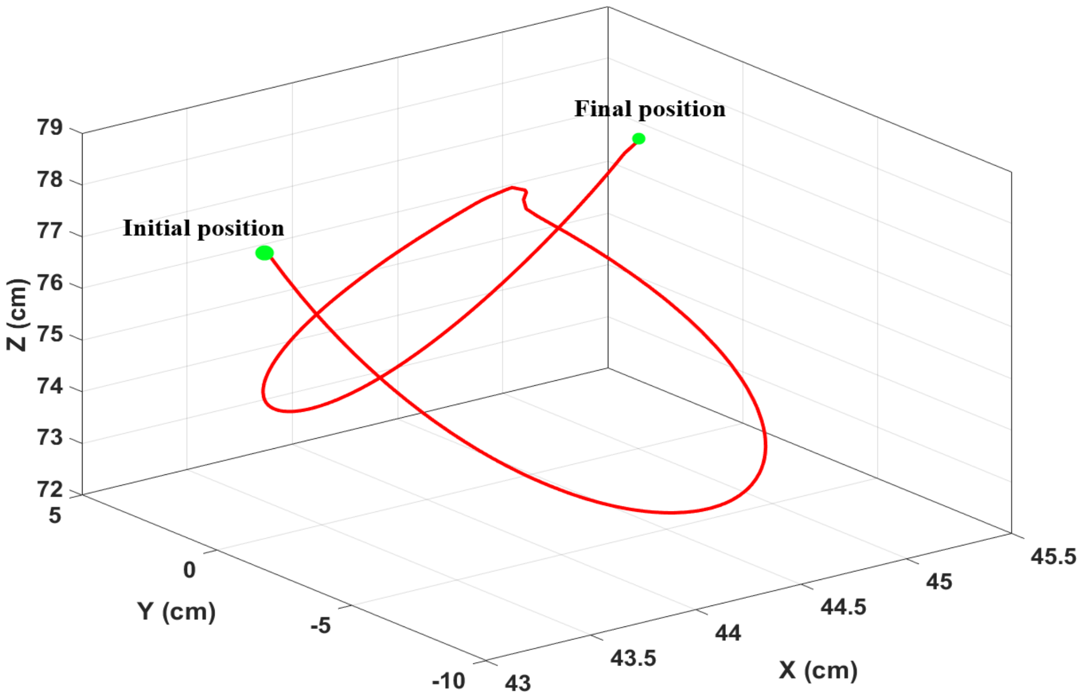


Figure 5.23: Path of the arm-base. This path is generated by making the legs in the same body-side move in similar movement phase.

Although, the conditions of this experiments are very hard, the response of the system is still admissible as shown in Figures 5.24, 5.25, and 5.26. It is worth to mention that, generating a path in this way makes the body weight goes suddenly to the right-hand side (when leg 1, 3, 5 in transfer phase), which causes legs slippage. In contrast, when legs 2, 4, and 6 are in transfer phase the robot's body is shifted in the left-hand side. Despite this fluctuation in the body, the arm retrieved its initial configuration, Figure 5.27.

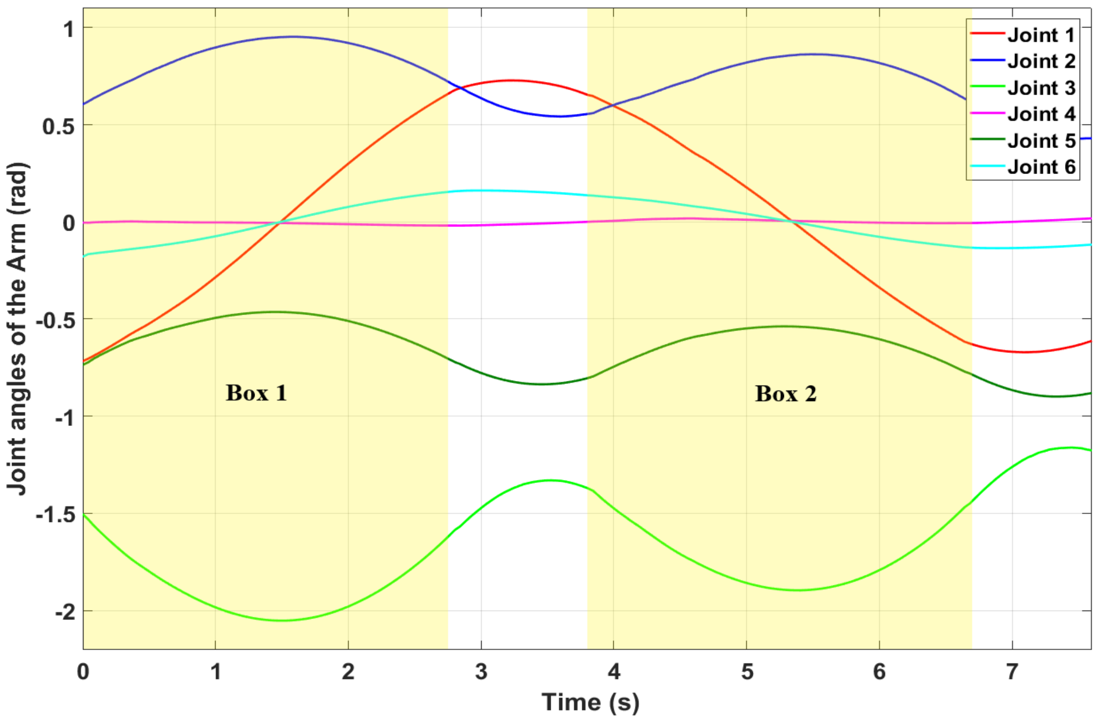


Figure 5.24: Arm joints angles, case of arbitrary base path.

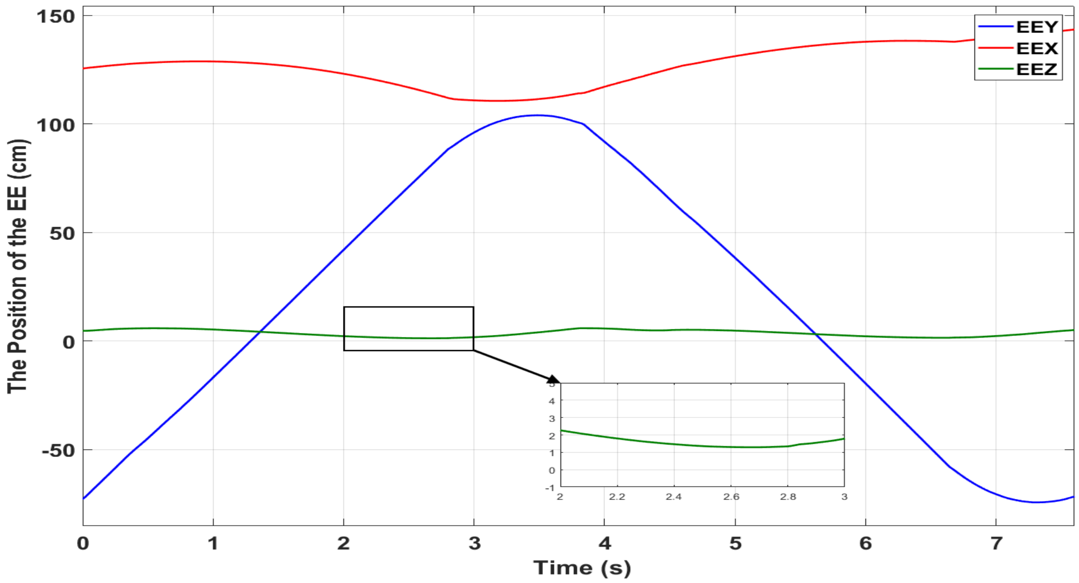


Figure 5.25: EE position in x, y, and z direction case of arbitrary base path.

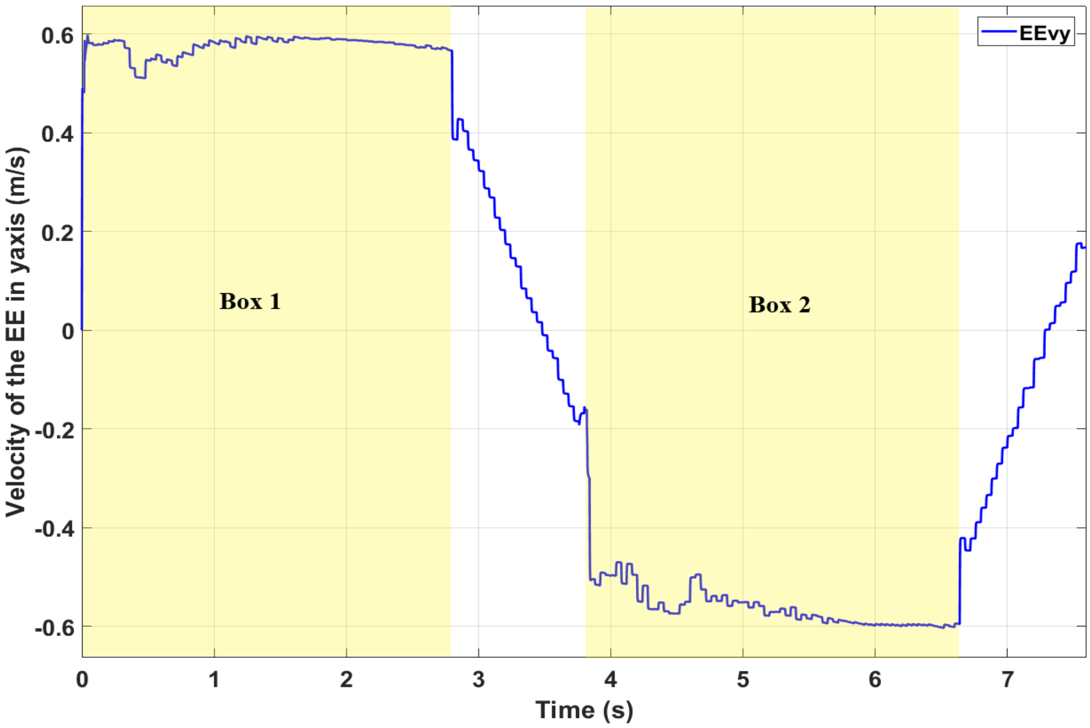
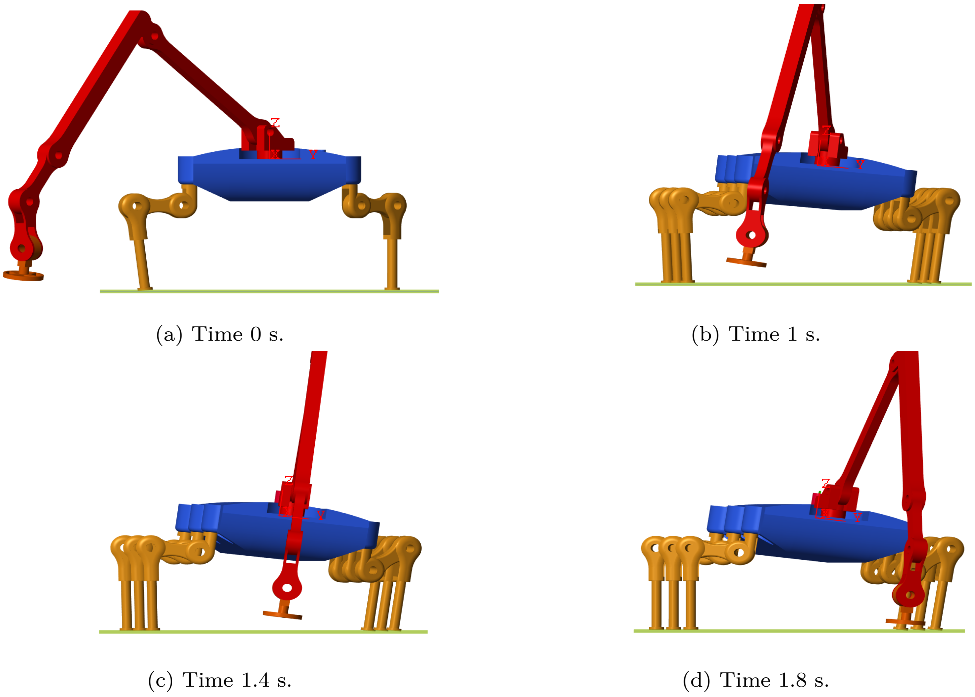


Figure 5.26: Velocity of the EE, case of arbitrary base path.



(a) Time 0 s.

(b) Time 1 s.

(c) Time 1.4 s.

(d) Time 1.8 s.

Figure 5.27: Performance of the arm during an arbitrary motion by the robot's body.

5.6 Summary

In this chapter, tracking a prescribed path in the operational-space by the EE in different arm-base position and orientation was presented. The path has been generated according to the constraints that are imposed by tasks, for instance the height of the path with respect to the ground and the dimensions of an object that may exist inside the path. The entire path is formed from a set of points and generated according to the initial point, height of the path, and the final point.

In order to verify the hypothesis, a continuous landmine-scanning manipulator-arm mounted on a six-legged robot was presented. A new trajectory planning has been introduced by generating a straight-line path for the sensor-head to transfer from side to side to scan a minefield using legged robot. The jerk effect due to the motion of the manipulator has been reduced, and the trajectories between adjacent boxes are blended by a semi-circular path.

Four experiments have been conducted to verify the efficiency of the controller and the trajectory planner. In the first experiment, the effectiveness of the manipulator to track a prescribed path with 5 cm in height above the ground was evaluated while the arm base followed a straight-line.

In the second experiment, the response of the arm to avoid an obstacle was evaluated. The path was modified according to the location and the dimensions of the object. The geometry of the object was extracted by cameras, which fixed on the robot's body. In spite of this change in the path shape, the EE motion rate was maintained. The role of the gravity was obvious to reduce the EE speed slightly, especially at the time of avoiding an obstacle.

In term of changing the arm-base position and orientation, two cases were considered. First, the arm-base was gradually dropped by 10 cm along z-axis. The response of the EE to follow a straight-line path was acceptable. Second, the ability of the EE to track a line path while the arm-base followed an arbitrary was conducted. Although a slippage at the robot legs has been encountered while generating this motion, the performance was acceptable. The purpose of the last two experiments was to investigate the effectiveness of the controller to cope with the variation of the robot body due to irregularity of the terrain.

CHAPTER 6

The Dynamics of the Robot with Arm

6.1 Introduction

In addition to the kinematic constraints, the functionality of legged robots is determined by the multi-body dynamic system and the contact forces. While walking the environment imposes sets of dynamic constraints on the robot's parts; hence, using absolute motion control schemes are insufficient. Unless otherwise, controlling the robot's motion task should be planned carefully; but, this requires a precise modelling for robot's kinematics, dynamics, and the geometry of the surrounding.

The method of inverse kinematics finds the joints angles of the robot by given Cartesian position and orientation in the task-space. In order to guarantee the stability criteria, the system should be dynamically constrained [127]. Many dynamic quantities are necessary to take into account for controlling the robot motion, such as, the normal and tangent forces at the contact points between the robot and its environment [91].

The equation of motion of any robotic system consists of three variables, namely, torque, force, and acceleration [55]. Although joints torques are only required as a control command, the other variables (accelerations and forces) should be computed [62]. Whether the solution is obtained analytically or numerically, there are different types of approaches to consider the dynamic constraints for designing a controller of legged robot. The first one is based on finding acceptable force at the contact point and then calculating joints torque and acceleration accordingly [166, 167]. The second approach is to get the required joint accelerations, then computing for joint torques, no need to include the contact force [64, 136]. Finally, the approaches that consider all variables in one optimisation problem at the same time [62, 168].

In the context of humanoid robot, Mistry et al. [64] proposed a method using QR decomposition to derive torque command without need to explicitly compute for contact force. While this method is appealing in controlling bipedal robots, it cannot handle more than two contacts in continuous motion [116]. Furthermore, they assumed that the velocity at the contacts is zero and this constraint should hold during the control cycle. However, this assumption is practically not feasible, especially when an external force is applied to the robot's body.

[82] designed an operational-space inverse dynamics to generate a whole body behaviour by decoupling the task space and the robot's dynamic. The method is based on projecting the tasks in the null pace of the constraints to ensure a proper decoupling between tasks and the constraints. Although, this approach can compute the inertia matrix implicitly, it involves much more computations.

The aforementioned methods can handle equality constraints; however, many constraints (especially caused by dynamic) can only represented as inequality constraints, such as, the contact forces.

Far from the analytical solutions using pseudo-inverse techniques, the problem of inverse dynamics can be formulated using QP to handle both inequality and equality constraints. Considering more than one task in a different level of priority is the obvious drawback of the classical QP. An interesting contribution was achieved by

[62] to consider all the optimisation variables at once in same optimisation cycle. The computation cost has been potentially reduced, and both type of constraints have been processed in any level of importance. This method is applied on a bipedal robot with more than two contacts; however, no mention to the transition of the constraints.

The main contribution of this chapter is twofold. The dynamics effects of the robot are considered as additional constraints to be satisfied, and the redundant space with respect to the operational tasks is optimally exploited to accomplish subsequent tasks.

In order to demonstrate the effectiveness of the approach, the performance of the robot is investigated in three scenarios. First, a comparison between a model based on use all force components and a model with just normal force. Second, the dynamic impact of the arm is tested with, without the arm, and with different weight of sensor-head. Finally, the effect of the ground irregularity is inspected.

6.2 Robot Dynamics Model

Figure 6.1 shows a six legged robot with three contact forces at each leg. The perpendicular component is in z direction and the other tangential forces in x and y directions are in the contact plane. The attitude of the robot is specified by the position and orientation of its CoG, and controlled by the position and orientation of the body and joints with respect to the inertial frame of reference.

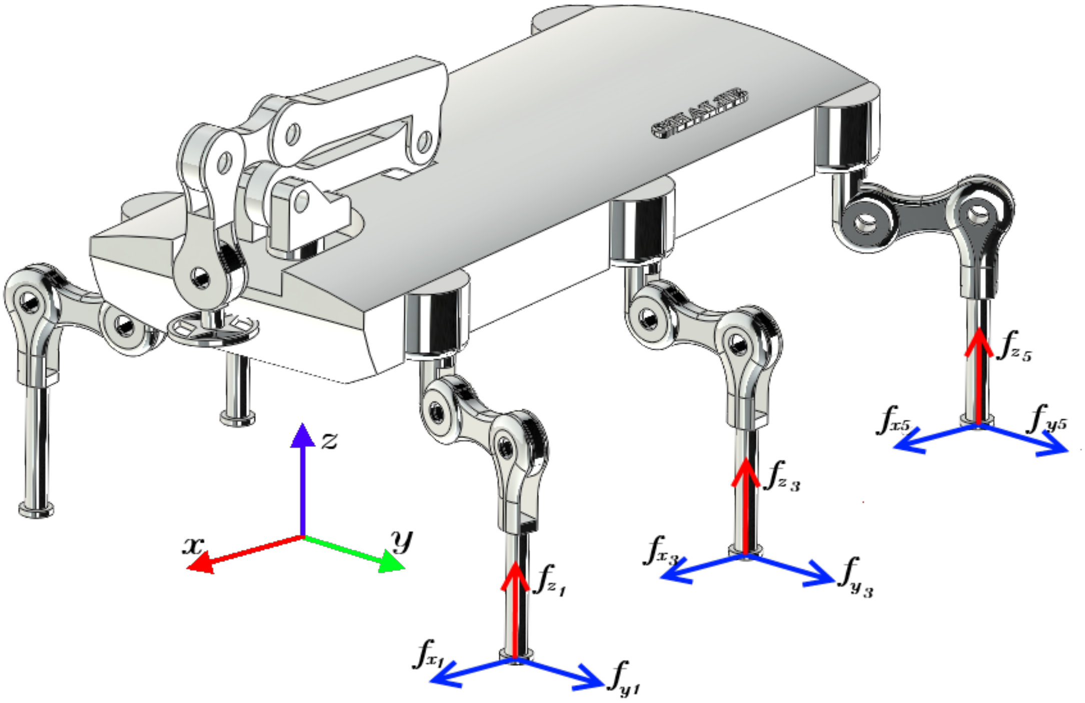


Figure 6.1: Force components at each leg.

$$M(q)\ddot{q} + H(q,\dot{q})\dot{q} + G(q) + J_c^{\intercal} f_c = S^{\intercal}\tau \tag{6.1}$$

where M is the $(6+n)\times(6+n)$, symmetric positive-definite generalised inertia matrix of the system, $H(q,\dot{q})\dot{q}$ is $(6+n)\times 1$ vector of Coriolis and centrifugal force, $G(q)$ is $(6+n)\times 1$ vector of gravity force, τ is the n vector of joint torque, $S = [I_{n\times n}\, 0_{n\times 6}]$ is the selection matrix, which describe the under-actuation, $J_c = \partial x_c/\partial q$ is the Jacobian matrix at the contact, and f_c are the contact forces. See appendix A for full derivation of the dynamic of the robot.

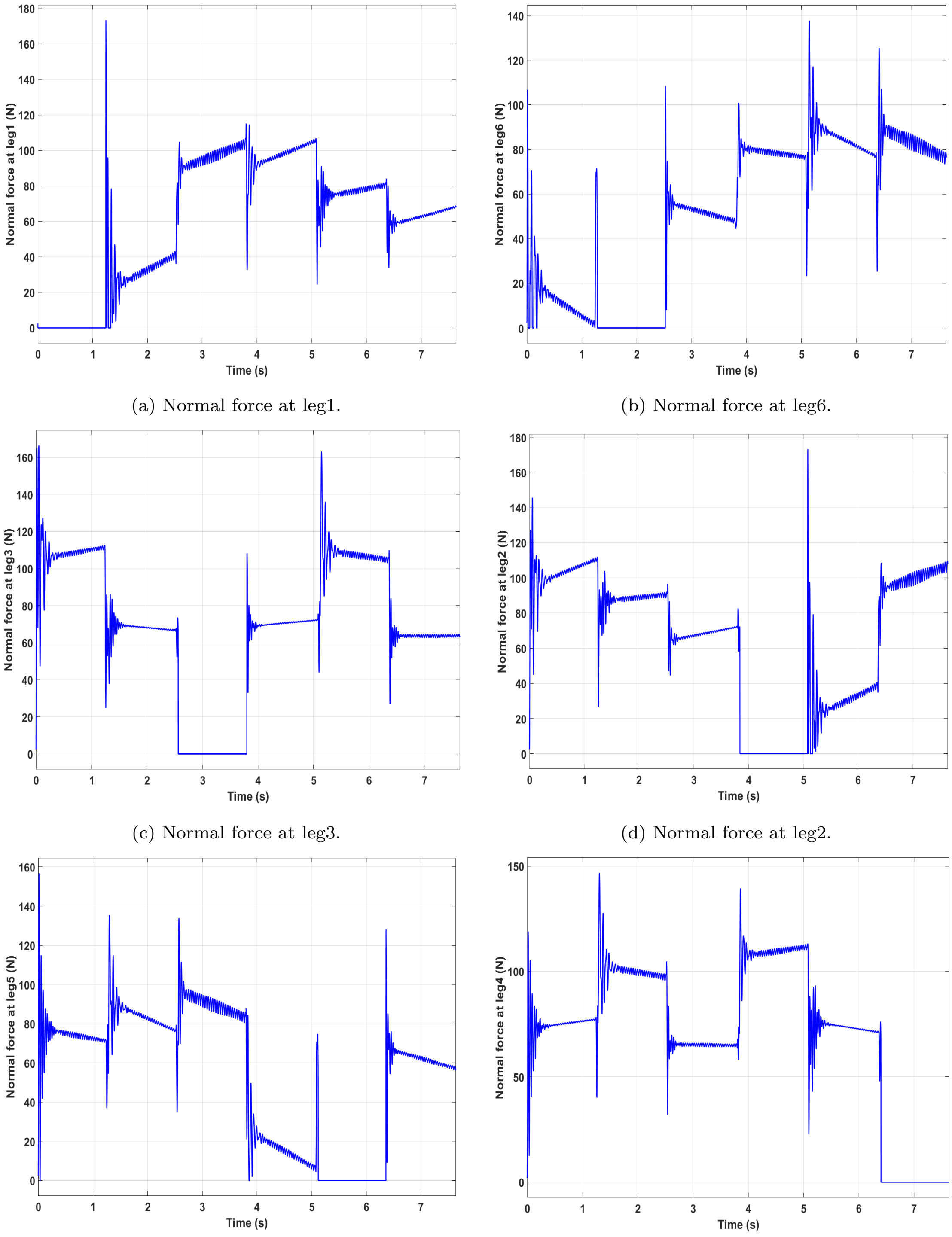
The dynamic model of the robot in rigid contact with the environment is represented by Equation (6.1). At the dynamic level, the reference behaviour is specified by the expected task acceleration $\ddot{x}$; and the control command is typically the joints torque τ. The operational space inverse dynamics refers to the problem of finding the torque control input τ that produces the desired acceleration at a task $\ddot{x}^*$, using any necessary joint acceleration $\ddot{q}$.

Two necessary conditions have to be satisfied at the contact point. The first one is related to the normal force should be greater than or equal to zero, and because the robot can push against the ground and cannot drag; this constraint is unilateral.

$$f_c^{\perp} \geq 0 \tag{6.2}$$

where $f_c^{\perp}$ are the normal elements of f_c.

Figure 6.2 represents the normal forces at legs (1, 6, 3, 2, 5, and 4). At each step time, one leg is in transfer phase the other legs are in contact with the ground. According to the legs position and state, each leg has a different values of the normal force. For instance, leg 6 located between two legs (5 and 4), which is in stance at time 0 - 1.2 s; hence the normal force is small between 20 - 0 N. When either leg 5 or leg 4 is in transfer phase, the value of normal force of leg 6 is larger around 60 - 80 N at time 5.2 - 7.6 s.



(a) Normal force at leg1.

(b) Normal force at leg6.

(c) Normal force at leg3.

(d) Normal force at leg2.

(e) Normal force at leg5.

(f) Normal force at leg4.

Figure 6.2: Normal force at legs (1, 6, 3, 2, 5 ,4). The figures has been ordered according to the legs sequence in one cycle.

$$\ddot{x} \geq 0 \tag{6.3}$$

$$\ddot{x} f_c^{\perp} = 0 \tag{6.4}$$

The second necessary condition is that the acceleration when the leg is taking off (transfer time) is greater or equal to zero, as indicated in Equation (6.3). Figure 6.3 shows the acceleration of leg 1 during one cycle, the acceleration values is maintained to zero value at time of contact between time 1.2 - 7.6 s. Furthermore, the impact of other legs with the ground is obvious; hence, the approach in [64] is not applicable in a robot with more than two legs in contact with ground, such as the six legged robot.

Both conditions are complement each other and never happen at the same time; hence, the mathematical representation of this condition is written in Equation 6.4.

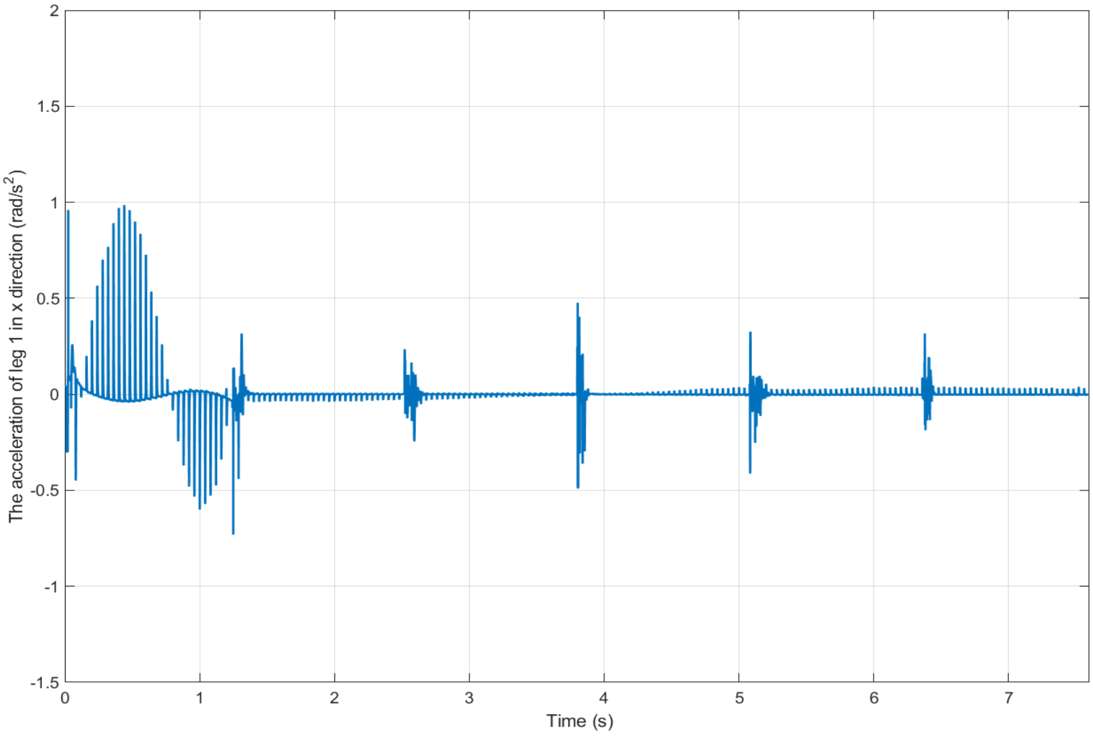


Figure 6.3: Acceleration of leg 1.

6.3 Methods to Handle Contact Forces

As previously stated, the zero velocity constraint, Equation (6.5), is widely used [98] in legged systems to ensure rigid contact with the ground. There are two reasons to consider this constraint. First, it is an equality constraint; hence, it is easy to implement in projection methods. Second, there is no need to include the contact force explicitly.

$$\dot{x}_c - J_c\dot{q} = 0 \tag{6.5}$$

where $\dot{x}_c$ is the velocity at the contact point, and $J_c = \dfrac{\partial \dot{x}_c}{\partial \dot{q}}$ is the Jacobian matrix of the contact point.

$$CQ^{\intercal}(M\ddot{q} + GF) = CQ^{\intercal}S^{\intercal}\tau + RF_c \tag{6.6}$$

$$UQ^{\intercal}(M\ddot{q} + GF) = UQ^{\intercal}S^{\intercal}\tau \tag{6.7}$$

where Q is an orthogonal matrix, R is an upper triangular matrix $C = [I_{k\times k}\, 0_{k\times(n+6-k)}]$, $U = [(0_{(n+6-k)\times k}\, I_{(n+6-k)\times(n+6-k)}]$, GF is the generalised forces (H and G in Equation (6.1)), F_c are the contact forces, and k is the number of constraints.

This method was adopted by [64], they decompose the dynamic equation into two equitations constrained and unconstrained, as shown in Equations (6.6) (constrained equation) and (6.7) (unconstrained equation). The aim of this decomposition is to decouple the constraints from the tasks. Then projecting the tasks in the null space of the constraints using the Moor-Penrose pseudo-inverse. As shown from Equation (6.7) the torque command is generated without including the contact forces.

Instead of eliminating the impact of the contact force, Sentis et.al. [82] proposed a method to compute the contact force from the equation of motion and Equation (6.5).

$$J_c\ddot{q} = -\dot{J}_c\dot{q} \tag{6.8}$$

$$\ddot{q} = -J_c^{-1}(\dot{J}_c\dot{q}) \tag{6.9}$$

$$f_c = (J_c^{\intercal})^{\dagger}(S^{\intercal}\tau - GF) + (J_c^{\intercal}M^{-1}J_c)^{-1}\dot{J}_c\dot{q} \tag{6.10}$$

where † denote to weighted pseudo-inverse.

This method is based on differentiating Equation (6.5) once and substituting for $\ddot{x}_c$ by 0 (according the zero velocity assumption) yields Equation (6.8). Solving Equation (6.8) for $\ddot{q}$ yields the acceleration in joint space in term of task-space acceleration as in Equation (6.9).

Multiplying Equation (6.1) by J_cM^{-1} and inserting Equation (6.9) the contact force can be deduced as in Equation (6.10).

The control law can be realised by re-injection Equation (6.10) into Equation (6.1). According to this method the joint space acceleration is omitted and replaced by the acceleration at the task space. The contact forces can only be represented as inequality constraints; therefore, the projection methods cannot handle them directly.

In the following section, the contact forces are formulated as inequality constraints and considered with other constraints in one optimisation problem.

6.4 QP in a Cascade

In order to implement sophisticated behaviours, a legged robot requires controlling many tasks at the same time. These tasks should be fully satisfied using the generalised motion space; otherwise, a proper decoupling between tasks must be implemented by imposing a strict hierarchy between them.

Describing the overall behaviour requires defining all task points and ordering them in a certain priority level. Each task point can be described by its position and orientation with respect to reference frame. The desired motion is achieved by controlling each point to realise a certain objective.

The Equations (6.1), (6.2), (6.3), and (6.8) represent the main constraints to be fulfilled. Following the idea of [62, 169], both type of constraint and tasks can be formulated as a hierarchy QP.

$$\min_{\tau,\ddot{q},f} \parallel \Phi \parallel \tag{6.11}$$

$$\ddot{x} = J_c\ddot{q} + \dot{J}_c\dot{q} \tag{6.12}$$

The optimisation variables $\Phi = (\tau, \ddot{q}, f_c^{\perp})$ is shown in Equation (6.11). Any task can be represented in acceleration level in the operational-space as in Equation (6.12).

In order to formulate the control law as cascade of QP, all the constraints are written as affine functions; In addition to the above constraints, the friction cone is defined around each leg to ensure the reaction force remain inside this cone. The friction cone is approximated to pyramids to get linear inequality constraints [170]

Equation (6.1) is given high priority to ensure that the generated motion is dynamically consistent. In order to guarantee there is no motion at the contact, the second constraint in Equation (6.5) is defined. All contact constraints in Equation (6.2) are satisfied before any task does not include contact force as in Equation (6.12). The constraints can be written in a lexicographic order[1] as (6.1) $\prec$ (6.5) $\prec$ (6.2) $\prec$ (6.12).

Generally, six legs robot requires at least three non adjacent legs on ground at any time to ensure the stability. In this work, the case of five legs in contact will be be considered. While increasing the number of legs will enhance the overall balance of the robot, the corresponding force variables will be increased. Consequently, the cost of the computation will be increased. Hence, extract the normal force components from the contact forces will reduce the number optimized variables [62].

6.5 Experiments

To verify that including the contact forces variables in optimisation problem will enhance the overall performance of the robot, three experiments have been conducted. In the first experiment, the robot will walk over flat ground, and the effect of decomposing the contact forces to normal force only is investigated. In order to test the robot's ability to continue walking despite obstacles, the robot will ride an object with height of 5 cm in second experiment. The effect of the manipulator arm dynamic is evaluated in the third experiment. The simulation elements are listed in Table 6.1.

[1] In optimisation, lexecographic order means that any decreasing in the cost of a task must not lead to increase the cost of tasks with lower priority level [171].

Table 6.1: The values of the simulation.

Contact stiffness (N/m)	200000
Contact damping (N/(m/s))	300
Joint stiffness (m.N/rad)	10
Joint damping (m.s.N/rad)	15
Ground damping (N/(m/s))	10
Ground stiffness (N/m)	10
Coefficient of kinetic friction	0.1
Coefficient of static friction	0.2
Integration scheme	ode15s
Control frequency (Hz)	1000
Robot DoF	18
CPU (GHz)	3.6

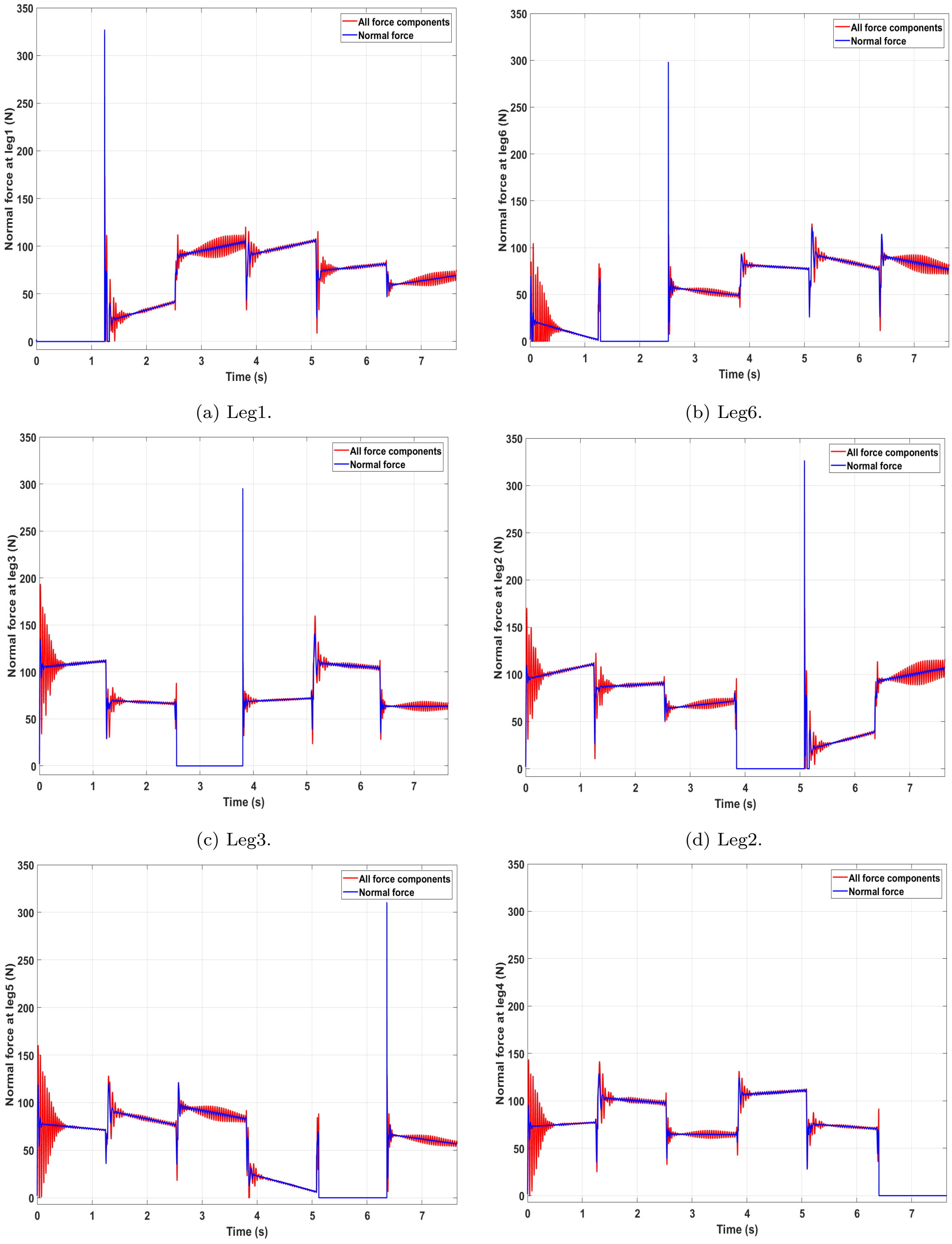
Table 6.2: The Physical Parameters of the Robot.

Part	Weight	Moment of inertia	CoM (cm)
	(kg)	$I_{xx}\, I_{yy}\, I_{zz}$ $(kg.m^2)$	
Hip	1	0.101, 0.89, 0.12	-2.62, -1.42, 2.5
Tibia	1	0.334, 0.463, 0.471	0.5, 0, 1.1
Ankle	1	0.363, 0.287, 0.287	13.3, 0, 0
Body	20	17.35, 11.77, 4.87	-2.4, -5.2, -0.04
Arm link1	0.5	0.001, 0.00, 0.000	7.2e-11, -0.0, -0.0
Arm link2	1	0.000, 0.044, 0.045	0.1, 4.9e-19, 0.0
Arm link3	1	0.000, 0.035, 0.036	0.04, -0.000, -2.23
Arm link4	0.5	0.035, 1.14382e-19, 0.005	0.035, 1.14e-19, 0.005
Arm link5	0.5	0.00, 0.00, 0.00	0.02, 0.001, 0.00
Arm link6	0.5	6.0e-19, 1.5e-18, -1.4e-18	0.013, 0.00, -0.00

6.5.1 Walking Task

In this experiment, two models have been considered to perform a normal walking task by the robot. In the first model, all force components were included in the optimisation problem. The force components have been decomposed to only normal forces in the second model. The results are presented as a comparison between the two models. The measured normal forces for both models are shown in Figure 6.4. A chattering at the contact point is obvious due to the instability in the solver in the first model. This phenomenon revels that decomposing the force components and taking the normal forces are necessary especially with point contact model.

The constraints of keeping the contact force inside the friction cone was satisfied by reducing tangential forces. As a consequent of reducing tangential force the corresponding friction forces are increased. Figure 6.5 illustrates the effect of the robot dynamics and the impact of the contact at other legs on the friction force of legs (1, 6, 3, 2, 5, 4). The value of friction force is significantly reduced in the second model. Although, friction forces are very necessary after switching from transfer to stance, as legs velocity transferred from certain a mount to zero. The friction force at leg one at the transfer phase, time 0 -1.2 s, is zero as there is no contact. At the moment of contact, this force instantaneously increased to contribute propelling the robot forward. The performance of the second model is better to keep the foot-tip inside the friction cone. This is clear particularly at the contact of the other legs time (2.6, 3.9, 5.1, 6.4) s. The same scenario was occurred to other legs, any variation in the friction values is because of the position of a certain leg. The video of this experiment is available at [172].



(a) Leg1.

(b) Leg6.

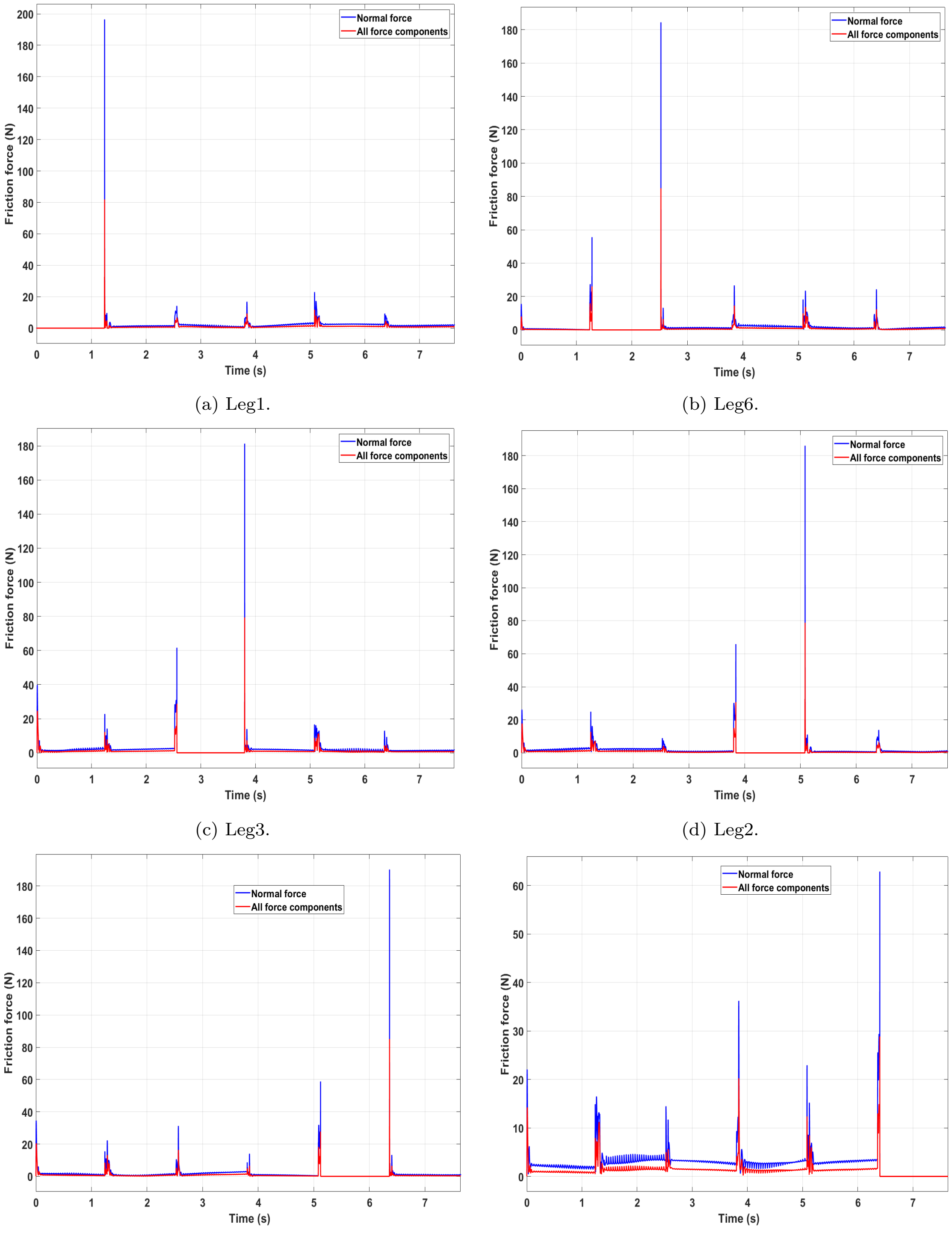
(c) Leg3.

(d) Leg2.

(e) Leg5.

(f) Leg4.

Figure 6.4: Normal force at the robot feet.



(a) Leg1.

(b) Leg6.

(c) Leg3.

(d) Leg2.

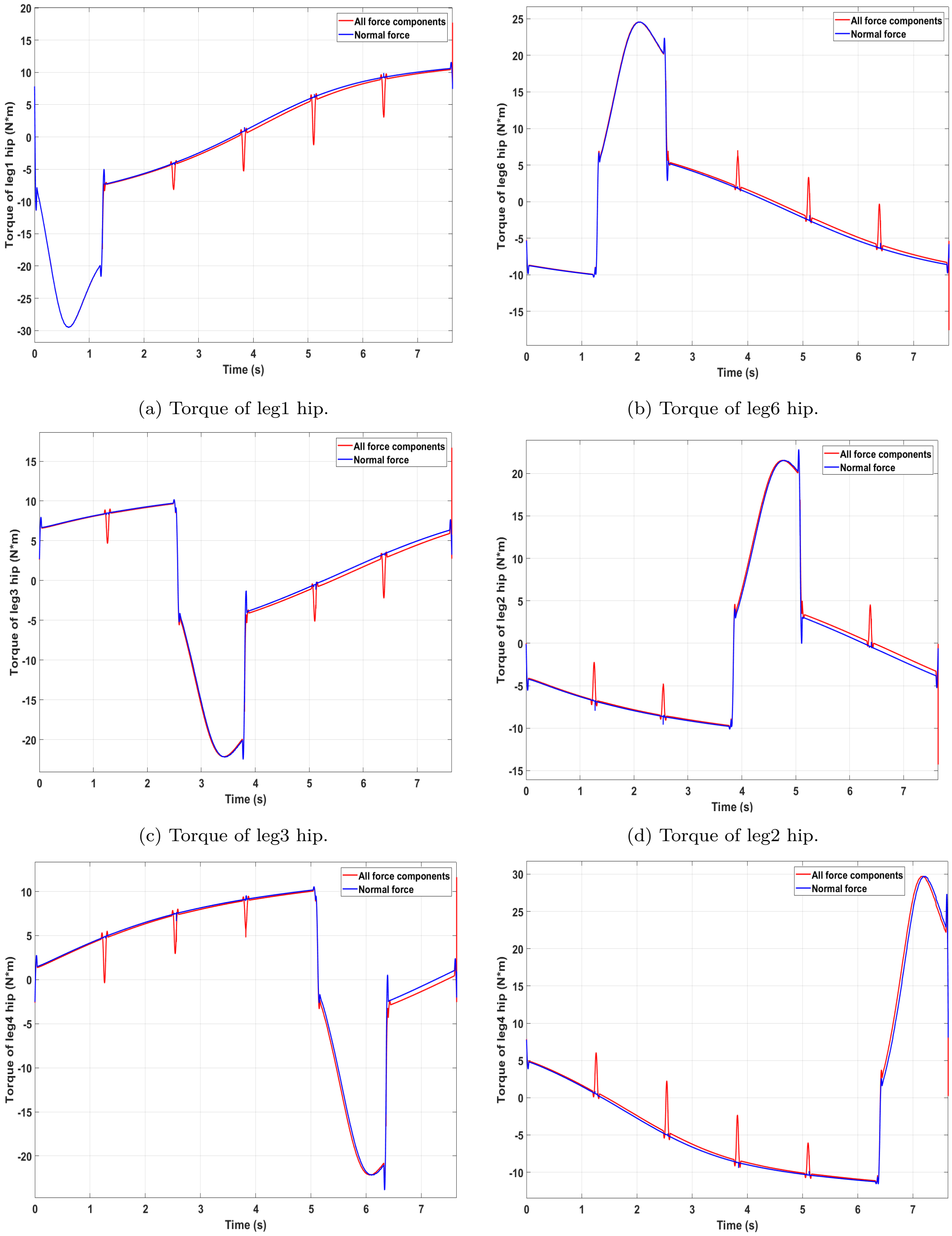
(e) Leg5.

(f) Leg4.

Figure 6.5: Friction force at the robot feet.

Although, the dynamic consistency constraints was satisfied in the first model, a discontinuity still appear in the torque command. Figure 6.6 shows the torque command of hip joints of legs (1, 6, 3, 2, 5, 4). In the Figure 6.6a, the exerted torque between time 0 - 1.2 s is exploited to transfer leg one from the initial position forward to landing point. Since there is no contact included in this period, the generated torque in both models is continuous. On the other hand, during stance phase, the discontinuity in the produced torque between time 1.2 - 7.6 s is clear in the first model. This discontinuities were vanished in the second model.

Figure 6.7 demonstrates the velocity of legs feet-tip along x direction. During the transition phase (time 0 - 1.2 s), the velocity has risen from zero to around 0.28 m/s and then back to zero. While stance phase, the velocity constraints (velocity is zero at contact) has been violated at any time the legs make contact with the ground. The performance of the second model was better.



(a) Torque of leg1 hip.

(b) Torque of leg6 hip.

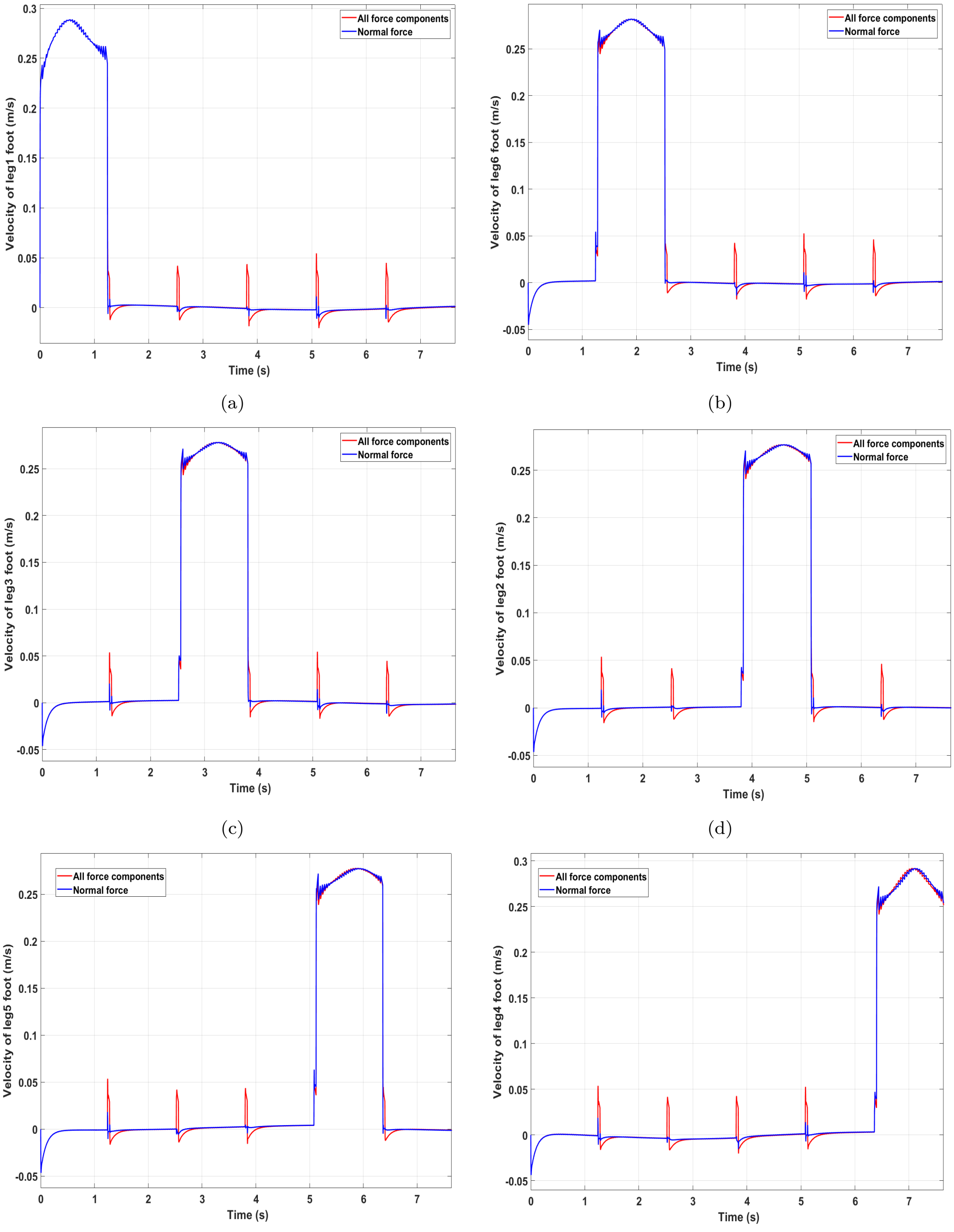
(c) Torque of leg3 hip.

(d) Torque of leg2 hip.

(e) Torque of leg5 hip.

(f) Torque of leg4 hip.

Figure 6.6: Torques of the legs joints.



(a) (b)

(c) (d)

(e) (f)

Figure 6.7: Velocities at the robot feet.

The overall velocity of the robot along x direction is illustrated in Figure 6.8. The robot walk in a constants speed of 0.044 m/s. A significant difference in the robot velocity was achieved by considering the second model.

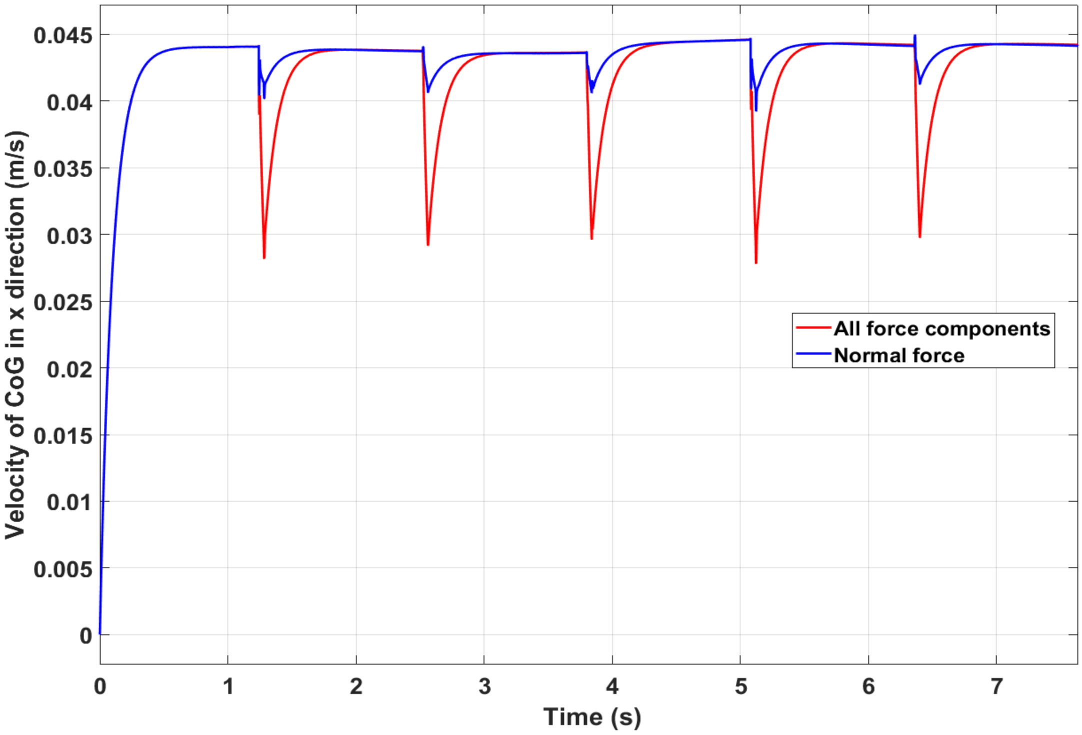


Figure 6.8: The overall velocity of the robot in x direction. Although, both results of the two models are appealing as neither velocities not became zero, the performance of the second model was better.

6.5.2 Stepping Over a Box while Walking

In this experiments, the ability of the robot to ride over an object and recovering the initial configuration are investigated. Figure 6.9 illustrates a hexapod robot walking for two cycle. Leg one experienced a box with height of 5 cm before the end of cycle one. Although, leg one has started cycle two from a different height, neither the robot velocity nor the body level were affected. Consequently, the controller exhibit a good compliant reaction.

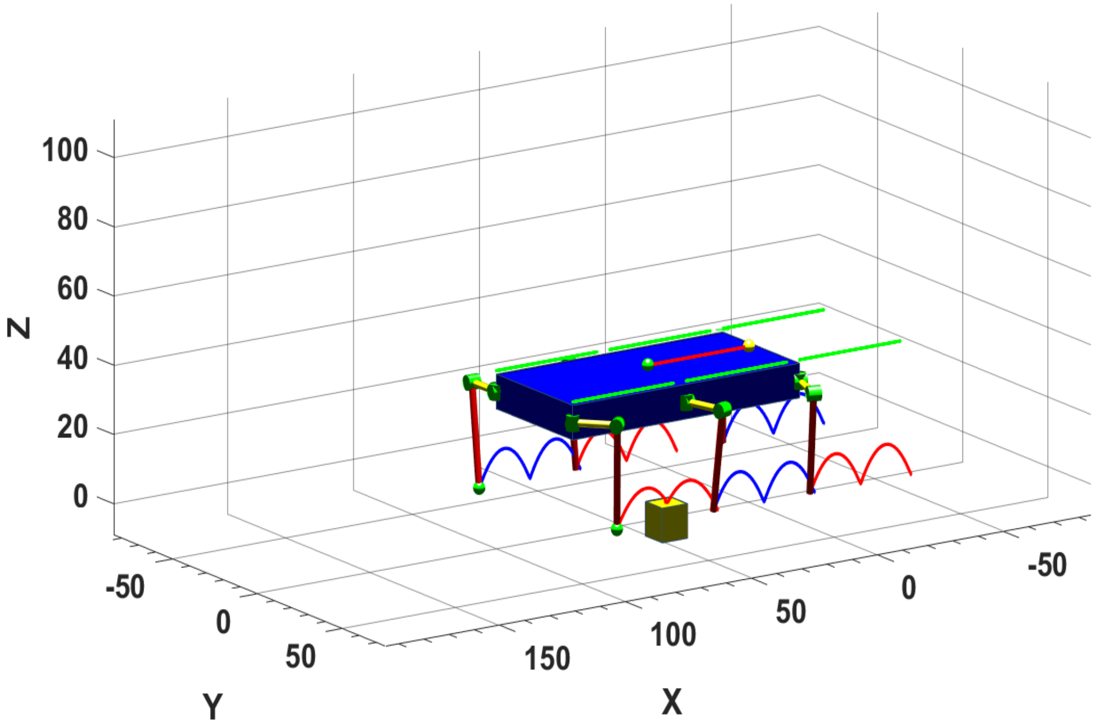


Figure 6.9: The robot perform two cycle walking. The robot traverse a distance of 40 cm through these two cycle. Both the continuity walking and getting the initial configuration back are satisfied. The legs 1, 4, and 5 path have traced by red line. Legs 2, 3, and 6 have traced by blue line.

A comparison between the leg one configuration in two cases is conducted. Case one when the robot walk over flat ground and when the robot stepping over a box in case two. The results of two cases is demonstrated in Figure 6.10.

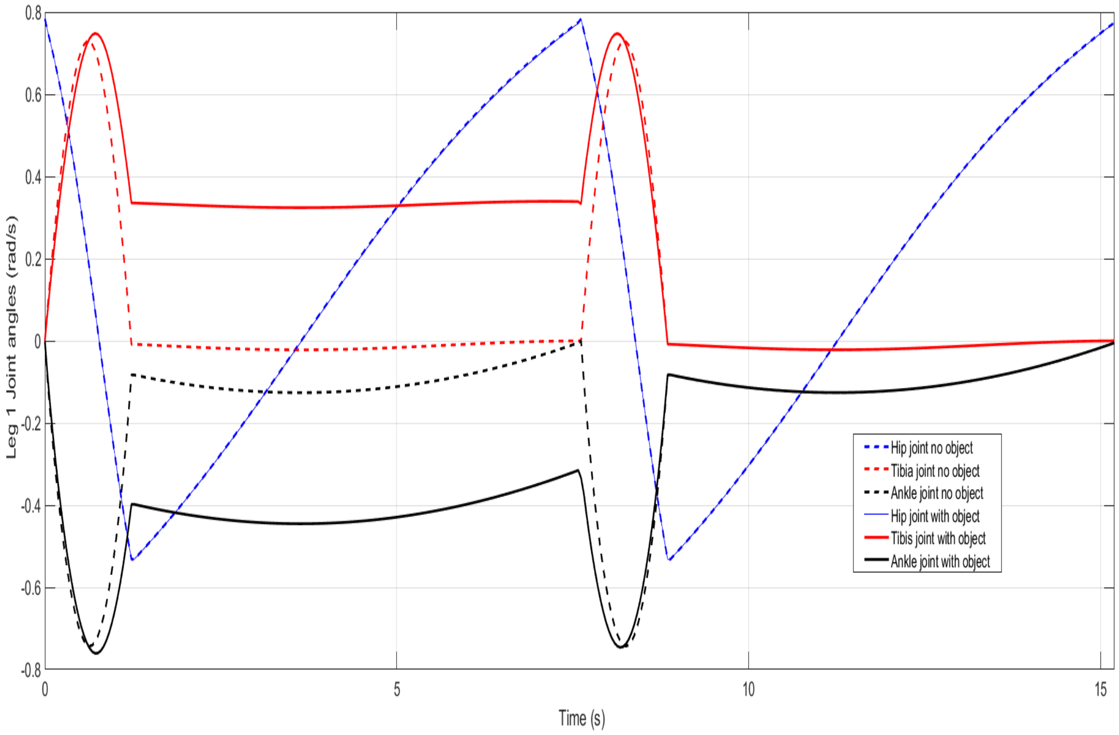
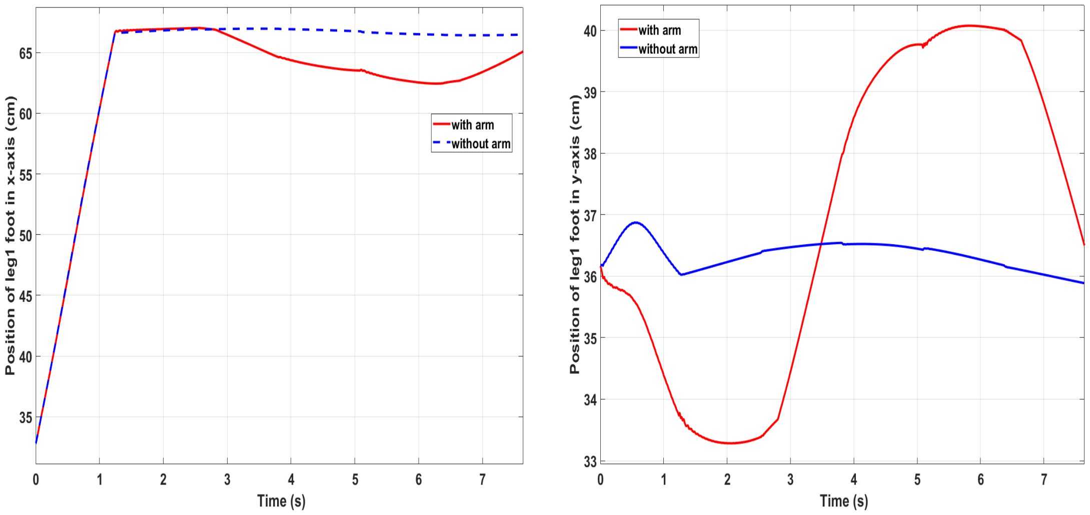


Figure 6.10: Two joint configuration of leg one. The first case when the robot walk over flat ground is indicated by dashed line. The second case when the robot rides a box with height of 5 cm while walking. While hip joint remain unchanged, both tibia joint and ankle joint took the responsibility to compensate for the presence of the box. After starting cycle two, specifically at time 8.2 s, the angles of both cases are aligned again as the situation is back.

6.6 The Dynamic Effect of the Arm

During the scanning operation, the manipulator arm has significant perturbations on the robot motion and stability. This impact is anticipated due the motion of the arm, which has a total weight of 4 kg and a total length of 190 cm, in an average speed of 0.5 m/s. During the operation mode, the average distance between the sensor-head and the front-edge of the robot is about 75 cm. In order to highlight on the robot parts and the control parameter that affected by the motion of the arm, a comparison between the robot with and without the arm has been conducted.



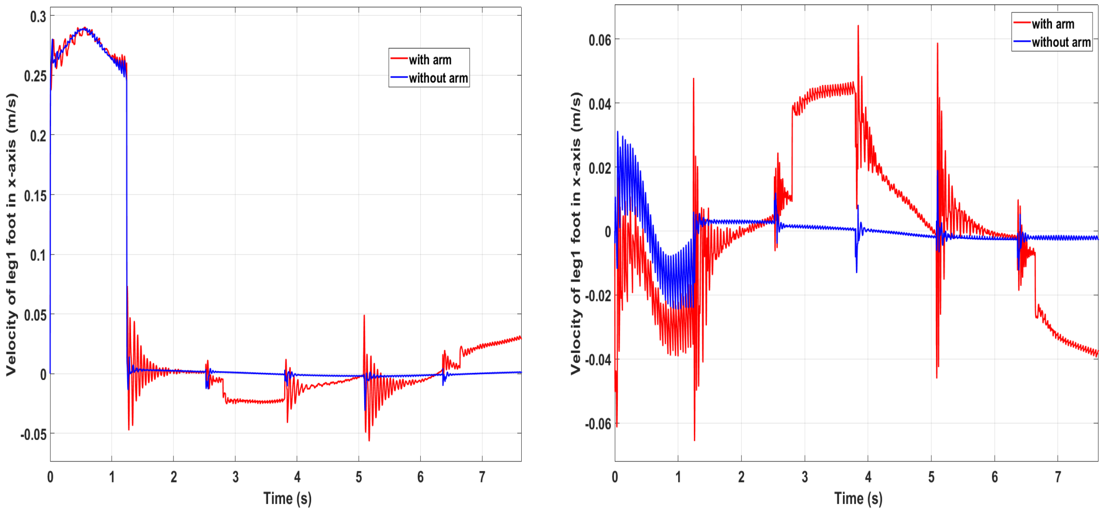
(a) Position of leg1 foot along x-axis.

(b) Position of leg1 foot along y-axis

Figure 6.11: Position of leg1 foot during one walking cycle. The blue line represents the leg1 position when the robot walk for one cycle without arm. The position of leg1 while using the arm is indicated by red curve.

As shown in Figure 6.11a, the position of leg1 foot along x-axis has been slightly drifted by 4 cm. Naturally, the arm tries to pull the robot body to the right when moving from right to left along y-axis and vice versa. Hence, its impact will be obvious on the position of the legs at the contact points in y direction. Figure 6.11b demonstrate the position of leg 1 has been deviated by 4 cm from the original position when the arm has not been fixed on the robot body.

The associated variation in the velocity at leg1 foot in x and y direction in illustrated in Figure 6.12. As it is clear from Figure 6.12b, the significant change in the velocity along y-axis at time 2.8 s when the arm tries to change its direction from right-left to left-right.



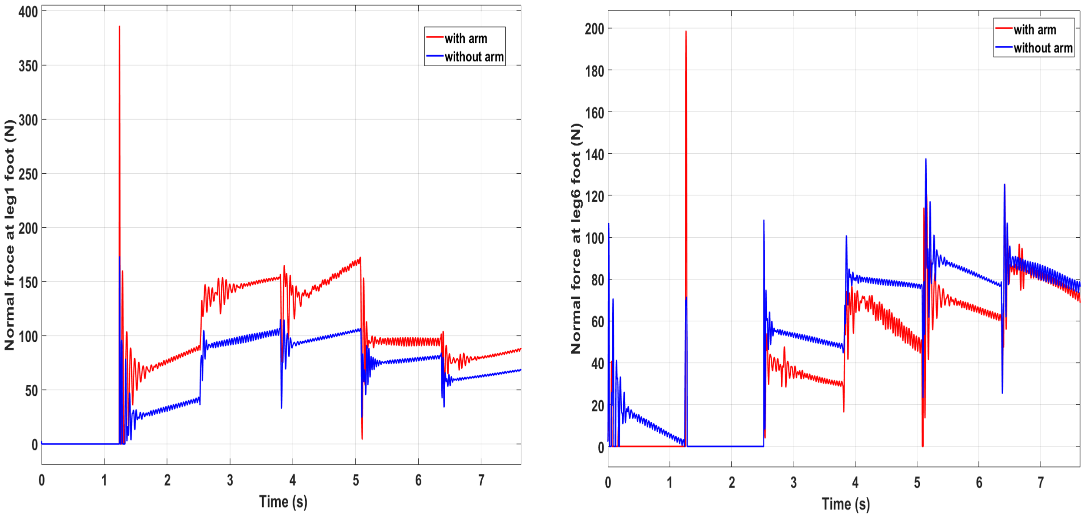
(a) Leg1 velocity in x direction.

(b) Leg1 velocity in y direction.

Figure 6.12: Leg 1 velocity in x and y direction. The blue line represents the leg 1 velocity when the robot walk for one cycle without arm. The velocity of leg 1 while using the arm is indicated by red curve.

Due to its position, the manipulator has a contradictory effect on the legs, which fixed in front of the body (leg 1 and leg 2), and the legs installed on the back (leg 5 and leg 6). For instance, the normal force at leg 1 will increase when adding the arm as shown in Figure 6.13a. On the other hand, the impact of the arm will reduce the normal force at leg 6, as shown in Figure 6.13b.

Intuitively to handle this phenomenon, the body's back could be designed in a weight bigger than the weight of the front part of the body. While this solution could work with statically stable body (no motion is involved), it cannot ensure the stability in quasi-static case. Therefore, distributing the body's weight between the legs by controlling the internal force would be a proper solution. This will be verified in this section. Further, the effectiveness of the controller will be tested by increasing the sensor-head weight to 2.5 kg.

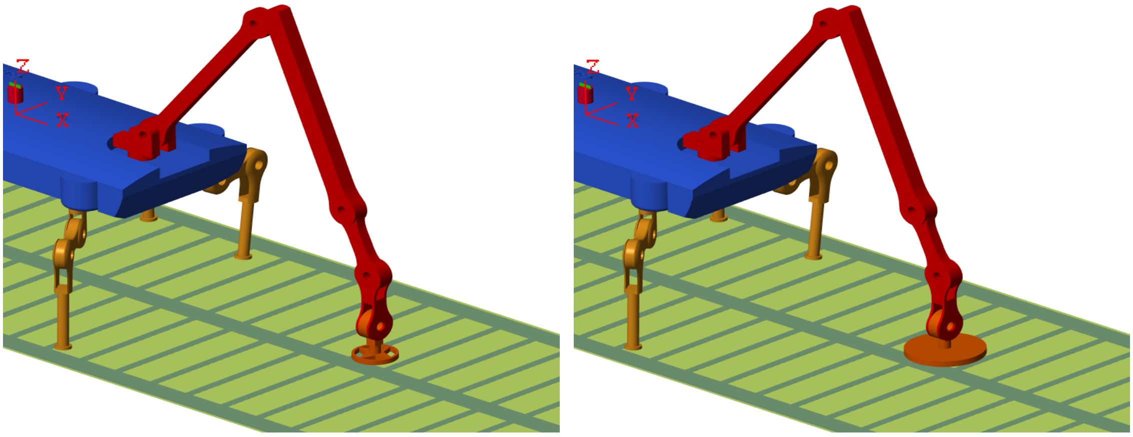


(a) Normal force at leg1.

(b) Normal force at leg6.

Figure 6.13: Normal force at leg 1 and leg 6 for two cases (with and without arm).

Figure 6.14 shows the arm of the robot with 0.5 kg sensor-head, Figure 6.14a and with 2.5 kg sensor-head, Figure 6.14b. In addition to these two cases, the case of the robot without the arm was considered to verify the effectiveness of the method.



(a) 0.5 kg sensor-head

(b) 2.5 kg sensor-head

Figure 6.14: Arm with different weight sensor-head.

Figure 6.15 shows three cases of the normal force at leg1. Case one, the robot perform a simple walking task without arm. This case has been chosen as reference for the other two cases. A sensor-head with two different weights of 0.5 kg and 2.5 kg were considered for the other two cases. The aim of this test was to check the ability of the robot to handle various type of detectors. As a result of force distribution between the legs in stance, the normal force at leg1 was increased between time 1.2 - 5.1 s, which is the period when the arm moved toward leg1. On the other hand, the contact force remained unchanged in three cases. This gives a perception that the role of leg1 was restricted at time 1.2 -5.1 s to compensate for the dynamic of the arm.

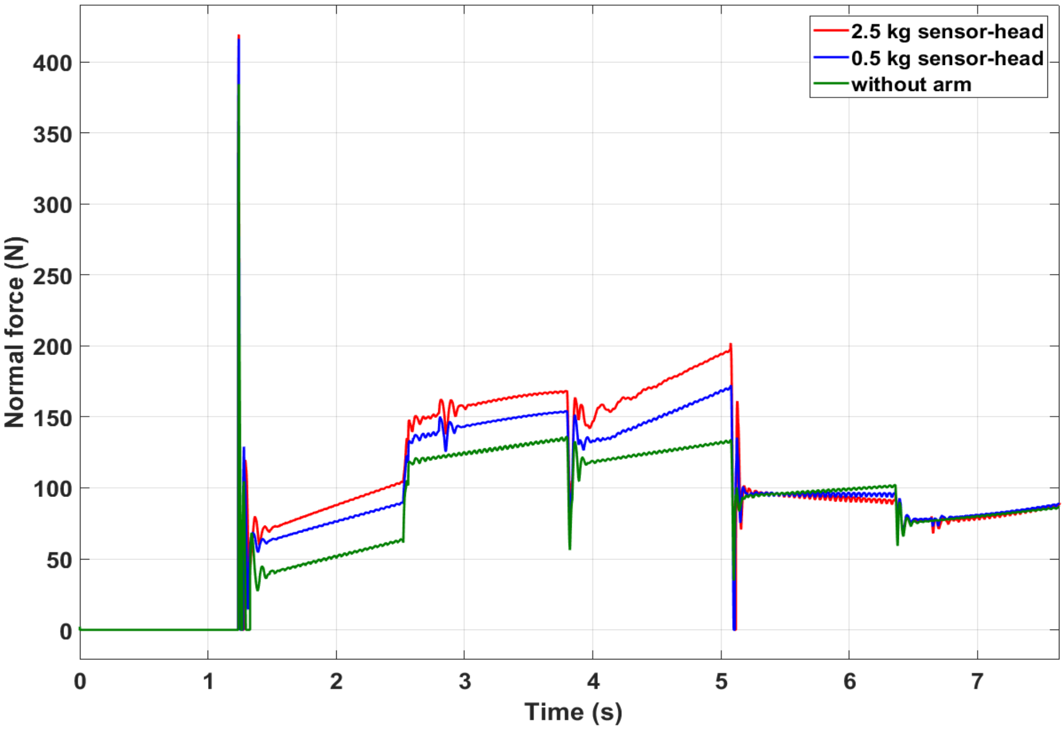


Figure 6.15: Normal force at leg 1 for three cases.

The situation of leg6 is different from leg1. If the results of Figure 6.13b are compared with the results of Figure 6.16, it is clear that impact of the arm has been reduced. The contact force in the second figure has been increased, this is explain the influence of the force distribution. Further, the values of forces stayed close for three cases between time 2.5 - 6.4 s. Variations have been occurred between the forces of the three cases at time 6.4 - 7.6 s, which is the time for lifting leg4 and time for the arm to return back to right-hand side. The videos of this experiment are available at [173, 174, 175].

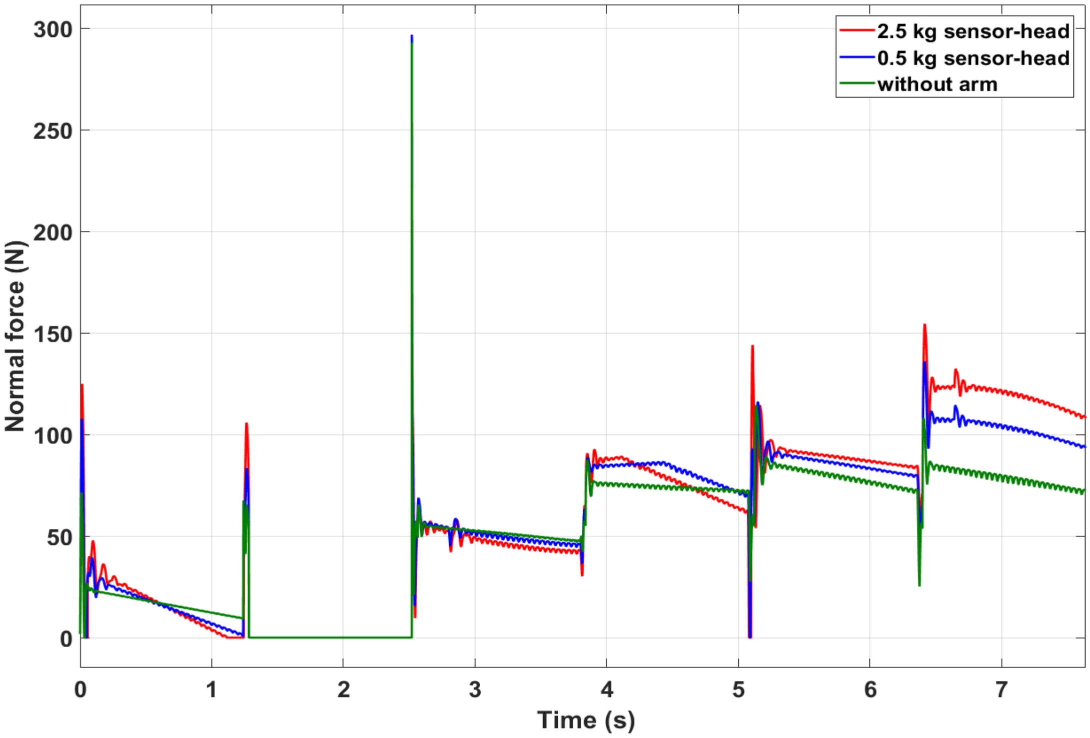


Figure 6.16: Normal force at leg6 for three cases.

6.7 Summary

In this chapter, the dynamic effects of the forces arisen from the interaction with the ground have been investigated, and the impact of the arm on the overall performance of the robot has been tasted and improved. The contact forces are formulated as linear inequality constraints. In order to reduce the problem size, the resulting contact constraints were decomposed to only the normal force. In addition to these constraints, some equality linear constraints, such as, the dynamic consistency and the zero velocity at the contact points have been included in the optimisation problem.

All these constraints are ordered in a cascade of linear equality and linear inequality and solved in QP solver. The null-space that arising from the joints redundancy has been used to control additional tasks.

The effectiveness of the controller was verified by conducting three experiments. In the first experiment, the robot performed a simple walking task over flat ground, and two models have been considered in this part, all force components have taken into account in the optimisation variables in the first model, and only normal forces are considered in the second model. The robot performance in terms of dynamic consistency and continuity has been significantly enhanced by the second model. The overall velocity of the robot body was improved by considering the normal and tangential forces arisen from the interaction between the robot and the environment.

In the second experiment, the effectiveness of the controller was verified by testing the ability of the robot to ride on an object. Both the continuity of the robot and recovering back the initial configuration are maintained during walking.

Finally, it was proven that holding the velocity equal to zero at the contact cannot be maintained throughout the process particularly when external forces applied on the robot. These forces represented by the motion the arm. The influence of the manipulator has been indicated on the position of leg 1 in x and y direction and compared with case of no arm. The same evaluation has been carried out on the normal force at leg1 and leg 6. These two legs have been chosen to study the effect of the arm on the front and back legs, and legs in both sides of the robot. The results revealed that adding more constraints such as the normal and translational force was necessary.

CHAPTER 7

Conclusions and Future Works

7.1 Conclusions

In this thesis, a hexapod robot with manipulator arm was introduced; both the design and the control aspects have been addressed. Due to many uncertainties that face the robot during achieving its work, controlling legged robots using traditional methods, such as position control, is not applicable. Dealing with the environment represents one of the most factors that generate these uncertainties. The six legged robot has at least three contact points with the ground (case of tripod gait). These contacts impose more constraints on the robot's motion and need to be satisfied before performing any task. Therefore, eliminating the effect of the contact forces using decomposition methods is insufficient in case of the six legged robot due to there are three or more contacts with the ground.

Formulating the tasks and the constraints as a linear function of equality and inequality is essential in the operational space control method. Although the traditional techniques, such as, projection methods can properly decouple tasks, but it cannot handle the inequality constraints. Stack of tasks method in multi-legged robot, such as hexapod robot, entails considering each constraints explicitly. Cascading QP method provides an efficient technique to handle inequality constraints in any priority level and imposing a strict decoupling between tasks; however, this method is computationally intensive. Hence, reducing some optimisation variables is very important in enhancing the overall performance of the robot.

The continuous walking has been achieved by tracking a specified reference path that defined at the robot's CoG. Although there are dips in the velocity curve due to the impact of the contact between the robot and the ground, these dips are occurred for very short time about (0.1 s) and the overall speed of the robot did not reach to zero (about 0.012 m/s). The motion continuity of the robot has been improved significantly by considering the contact forces components in the optimisation problem. The spikes in the velocity curve during the legs contact with the ground are reduced by 63% from the average dropping at each contact. In spite of the rotation of the robot's feet during propelling the body forward, the slippage range is acceptable. Table 7.1 summarises the continuity enhancement of the robot's velocity by defining additional constraints.

Table 7.1: The continuity enhancement with different constraints.

The constraint	The robot velocity (m/s)	Dips average
Zero velocity	0.044	0.028
All force components	0.044	0.016
Normal force only	0.044	0.004

The continuity of the manipulator has been improved by defining the path at the sensor-head a straight line instead of a semi circular path that is used in the traditional method, and the effect of the jerk has been reduced significantly.

There are mutual dynamic effects between the robot body and the arm. The impact of the body on the arm, which discussed in Chapter four, was compensated by the joints of the arm. On the other hand, the dynamic effect of the arm due to its motion from side to other side, which presented in Chapter six, was treated by controlling the internal forces of the robot. The slippage range along y-axis was enhanced by 90% if it compared with the position control methods.

Ultimately, having achieved the continuity motion in multi-legged robots in uneven ground and in presence of external forces will open up the use of these robots in applications that required quasi-static or dynamic balance rather than static balance.

7.2 Future Works

Although there is a vast of research in the field of controlling and designing of legged robots, new challenges will arise when assigning a specific task to a legged robot. It is necessary to handle these challenges effectively using efficient algorithms to get high performance robot. The suggestions and recommendations for future works are listed in the following:

- Some constraints, such as friction cone, have been represented in this work as linear inequality constraints. However, considering the non-linear form of these constraints could improve the overall performance of the robot.
- Due to its direct correlation in the motion of the robot, improving the trajectory smoothing using efficient methods, such as, reinforcement learning algorithms will be very important. Trajectory generation has a significant role in the smooth transition of the body parts and reducing the impact of the interaction between the robot and the ground.
- During walking the ground reaction is indispensable not only at the contact points, but also on the other part of the robot. Therefore, defining a zero

moment point (ZMP) at a certain point on the robot body, such as CoM, might provide a proper strategy to handle the dynamics associated with the interaction.

- Although there are many simulators that offers simulating many dynamic effects, applying the proposed approach using a real six legged robot would be very necessary to handle unexpected situations, when they arise.

Bibliography

[1] BostonDynamics, “SpotMini,” 2011. [Online]. Available: https://www.bostondynamics.com/

[2] P. G. D. Santos, E. Garcia, J. Estremera, M. A. Armada, C. C. Real, and L. Poveda, “DYLEMA : Using walking robots for landmine detection and location,” *International Journal of Systems Science*, vol. 36, no. 9, pp. 545–558, 2005.

[3] E. Colon, G. De Cubber, H. Ping, J. C. Habumuremyi, H. Sahli, and Y. Baudoin, “Integrated robotic systems for humanitarian demining,” *International Journal of Advanced Robotic Systems*, vol. 4, no. 2, pp. 219–228, 2007.

[4] P. Gonzalez de Santos, J. A. Cobano, E. Garcia, J. Estremera, and M. A. Armada, “A six-legged robot-based system for humanitarian demining missions,” *Mechatronics*, vol. 17, no. 8, pp. 417–430, 2007.

[5] L. Marques, a. T. D. Almeida, M. Armada, R. Fernández, H. Montes, P. González, and Y. Baudoin, “State of the Art Review on Mobile Robots and Manipulators for Humanitarian Demining,” *IARP WS on Humanitarian Demining*, pp. 2–8, 2012.

[6] K. Nonami, S. Masunaga, D. Waterman, H. Aoyama, Y. Takada, S. Masunaga, and Y. Takada, “Mine Detection Robot and Related Technologies for Humanitarian Demining,” in *Humanitarian Demining.* INTECH Open Access Publisher, 2008, ch. 10.

[7] R. Ponticelli, E. Garcia, P. G. de Santos, and M. Armada, “A scanning

robotic system for humanitarian de-mining activities," *Industrial Robot: An International Journal*, vol. 35, no. 2, pp. 133–142, 2008.

[8] M. K. Habib, "Humanitarian demining: Difficulties, needs and the prospect of technology," in *IEEE International Conference on Mechatronics and Automation*, 2008, pp. 213–218.

[9] S. Masunaga and K. Nonami, "Controlled metal detector mounted on mine detection robot," *International Journal of Advanced Robotic Systems*, vol. 4, no. 2, pp. 237–245, 2007.

[10] S. S. Roy, A. K. Singh, D. K. Pratihar, and Others, "Analysis of Six-legged Walking Robots," in *Proceedings of the forteenth national conference on machine and mechanisms (NaCoMM), Durgapur, India*, 2009, pp. 259–265.

[11] Q. J. Huang and K. Nonami, "Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control," *Mechatronics*, vol. 13, no. 8-9, pp. 773–790, 2003.

[12] I. Doroftei and Y. Baudoin, "A concept of walking robot for humanitarian demining," *Industrial Robot: An International Journal*, vol. 39, no. 5, pp. 441–449, 2012.

[13] D. Belter and P. Skrzypczyński, "Posture optimization strategy for a statically stable robot traversing rough terrain," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 2204–2209.

[14] D. K. Pratihar, K. Deb, and A. Ghosh, "Optimal path and gait generations simultaneously of a six-legged robot using a GA-fuzzy approach," *Robotics and Autonomous Systems*, vol. 41, no. 1, pp. 1–20, 2002.

[15] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, "A reactive controller framework for quadrupedal locomotion on chal-

lenging terrain," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 2554–2561.

[16] R. Ponticelli and P. G. de Santos, "Obtaining terrain maps and obstacle contours for terrain-recognition tasks," *Mechatronics*, vol. 20, no. 2, pp. 236–250, 2010.

[17] M. Billah, M. Ahmed, and S. Farhana, "Walking Hexapod Robot in Disaster Recovery : Developing Algorithm for Terrain Negotiation and Navigation," *Engineering and Technology*, vol. 2, no. 6, pp. 328–333, 2008.

[18] M. Mistry, J. Nakanishi, and S. Schaal, "Task space control with prioritization for balance and locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 331–338.

[19] A. D. Prete, F. Nori, G. Metta, and L. Natale, "Motion-Force Control Prioritized Optimal Control Prioritized Motion-Force Control," in *IEEE International Conference on Robotics & Automation (ICRA).* IEEE, 2014, pp. 2540–2545.

[20] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *proceeding of 5th International Conference on Advanced Robotics.* IEEE, 1991, pp. 1211–1216.

[21] T. . T. Lee, C. M. . Liao, and T. K. Chen, "on the Stability Properties of Hexapod Tripod Gait." *IEEE journal of robotics and automation*, vol. 4, no. 4, pp. 427–434, 1988.

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

[23] F. Hardarson, "Stability analysis and synthesis of statically balanced walking for quadruped robots," Ph.D. dissertation, Royal Institute of Technology, KTH, 2002.

[24] J. Estremera, J. A. Cobano, P. G. de Santos, P. Gonzalez de Santos, P. G. de Santos, J. A. Cobano, and P. Gonzalez de Santos, "Continuous free-crab gaits for hexapod robots on a natural terrain with forbidden zones: An application to humanitarian demining," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 700–711, 2010.

[25] M. Hutter, C. Gehring, M. A. Höpflinger, M. Blösch, and R. Siegwart, "Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1427–1440, 2014.

[26] Yu Zheng and K. Yamane, "Human motion tracking control with strict contact force constraints for floating-base humanoid robots," in *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids).* IEEE, oct 2013, pp. 34–41.

[27] O. Khatib, L. Sentis, J. Park, and J. Warren, "Whole-Body Dynamic Behavior and Control of Human-Like Robots," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 29–43, 2004.

[28] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[29] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts," in *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 81–88.

[30] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the Rough-Terrain Quadruped Robot," in *IFAC Proceedings Volumes.* IFAC, 2008, pp. 10 822–10 825.

[31] E. Celaya and J. J. M. Porta, “A control structure for the locomotion of a legged robot on difficult terrain,” *IEEE Robotics \& Automation Magazine*, vol. 5, no. 2, pp. 43–51, jun 1998.

[32] Y. Yoshida, K. Takeuchi, Y. Miyamoto, D. Sato, and D. Nenchev, “Postural Balance Strategies in Response to Disturbances in the Frontal Plane and Their Implementation With a Humanoid Robot,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 6, pp. 692–704, jun 2014.

[33] M. Hirose and K. Ogawa, “Honda humanoid robots development,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, 2007.

[34] J. Montgomery, S. I. Roumeliotis, A. Johnson, and L. Matthies, “Human-in-the-loop Control of a Humanoid Robot for Disaster Response: A Report from the DARPA Robotics Challenge Trials,” *J. Field Robotics*, vol. 23, no. 00, pp. 245–267, 2006.

[35] H. Hirukawa, F. Kanehiro, K. Kaneko, S. Kajita, K. Fujiwara, Y. Kawai, F. Tomita, S. Hirai, K. Tanie, T. Isozumi, K. Akachi, T. Kawasaki, S. Ota, K. Yokoyama, H. Handa, Y. Fukase, J. ichiro Maeda, Y. Nakamura, S. Tachi, and H. Inoue, “Humanoid robotics platforms developed in HRP,” *Robotics and Autonomous Systems*, vol. 48, no. 4 SPEC. ISS., pp. 165–175, 2004.

[36] N. Pateromichelakis, A. Mazel, M. A. Hache, T. Koumpogiannis, R. Gelin, B. Maisonnier, and A. Berthoz, “Head-eyes system and gaze analysis of the humanoid robot Romeo,” in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1374–1379.

[37] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, “Design of the Hydraulically-Actuated, Torque-Controlled

Quadruped Robot HyQ2Max," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 635–646, 2017.

[38] M. Hutter, "StarlETH & Co-design and control of legged robots with compliant actuation," Ph.D. dissertation, ETH Zurich, Switzerland, 2013.

[39] J. M. Porta and E. Celaya, "Reactive free-gait generation to follow arbitrary trajectories with a hexapod robot," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 187–201, 2004.

[40] M. S. Erden and K. L., "Free gait generation with reinforcement learning for a six-legged robot," *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 199–212, 2008.

[41] S. S. Roy and D. K. Pratihar, "Dynamic modeling, stability and energy consumption analysis of a realistic six-legged walking robot," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 400–416, 2013.

[42] Y. Nonami, Kenzo and Huang, Qingjiu and Komizo, Daisuke and Fukao, Yoichiro and Asai, Yoshitomo and Shiraishi, Yoshinori and Fujimoto, Masaki and Ikedo, "Development and Control of Mine Detection Robot COMET-II and COMET-III," *JSME International Journal*, vol. 46, no. 3, pp. 881–890, 2003.

[43] G. Zhong, H. Deng, G. Xin, and H. Wang, "Dynamic Hybrid Control of a Hexapod Walking Robot: Experimental Verification," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2016.

[44] D. Krasny and D. Orin, "Generating High-Speed Dynamic Running Gaits in a Quadruped Robot Using an Evolutionary Search," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 4, pp. 1685–1696, aug 2004.

[45] M. S. Erden and K. Leblebicio, “Torque Distribution in a Six-Legged Robot,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 179–186, 2007.

[46] L. V. Bruno Siciliano, Lorenzo Sciavicco and G. Oriolo, *Robotics Modelling, Planning and Control.* Springer Science & Business Media, 2010.

[47] E. G. Pablo Gonzalez de Santos and J. Estremera, *Quadrupedal Locomotion An introduction to the control of Four-Legged Robots.* Springer Science \& Business Media, 2006.

[48] M. Vukobratović and B. Borovac, “Zero-Moment Point — Thirty Five Years of Its Life,” *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, 2004.

[49] S. Zhang, X. Rong, Y. Li, and B. Li, “A composite COG trajectory planning method for the quadruped robot walking on rough terrain,” *International Journal of Control and Automation*, vol. 8, no. 9, pp. 101–118, 2015.

[50] T. Sugihara, “Mobility Enhancement Control of Humanoid Robot based on Reaction Force Manipulation via Whole Body Motion,” Ph.D. dissertation, University of Tokyo, 2004.

[51] D. Guo and Y. Zhang, “Acceleration-Level Inequality-Based MAN Scheme for Obstacle Avoidance of Redundant Robot Manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6903–6914, dec 2014.

[52] W. Miller, “Real-time neural network control of a biped walking robot,” *IEEE Control Systems*, vol. 14, no. 1, pp. 41–48, feb 1994.

[53] J. C. Arevalo and E. Garcia, “Impedance control for legged robots: An insight into the concepts involved,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1400–1411, 2012.

[54] N. Hogan, “Impedance control: An approach to manipulation: Part II—Implementation,” *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 8–16, 1985.

[55] O. Khatib, “A unified approach for motion and force control of robotic manipulators,” *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[56] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal, “Operational Space Control: A Theoretical and Empirical Comparison,” *International Journal of Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.

[57] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, and R. Siegwart, “Quadrupedal locomotion using hierarchical operational space control,” *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1047–1062, 2014.

[58] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, “Task-Priority Based Redundancy Control of Robot Manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.

[59] P. Baerlocher and R. Boulic, “Task-Priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, no. October, pp. 323–329, 1998.

[60] J. Park and O. Khatib, “Contact consistent control framework for humanoid robots,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. May, pp. 1963–1969, 2006.

[61] L. Sentis, “Synthesis and Control of Whole-Body Behaviours in Humanoid Systems,” Ph.D. dissertation, Stanford University, 2007.

[62] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[63] M. Kalakrishnan, "Learning Objective Functions for Autonomous Motion Generation In Partial Fulfillment of the," Ph.D. dissertation, UNIVERSITY OF SOUTHERN CALIFORNIA, 2014.

[64] M. Mistry, J. Buchli, and S. Schaal, "Inverse dynamics control of floating base systems using orthogonal decomposition." in *ICRA*, 2010, pp. 3406–3412.

[65] J. E. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," *Fast Motions in Biomechanics and Robotics*, vol. 340, pp. 299–324, 2006.

[66] A. Dietrich, C. Ott, and A. Albu-Schäffer, "Multi-objective compliance control of redundant manipulators: Hierarchy, control, and stability," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 3043–3050.

[67] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Task-Based Robot Grasp Planning Using Probabilistic Inference," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 546–561, jun 2015.

[68] J. G. Cebada Reyes, F. F. Hahn Schlam, A. Ruiz Garcia, E. Romantchik Kriuchkova, and A. Michua Camarillo, "Design of a Position Control Based on Cuckoo Search Tuning for a Cutter Leaves Robot," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2085–2092, may 2016.

[69] Y. Liu and Y. Zhang, "Toward Welding Robot With Human Knowledge: A Remotely-Controlled Approach," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 769–774, apr 2015.

[70] C. A. Klein and S. Ahmed, “Repeatable pseudoinverse control for planar kinematically redundant manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 12, pp. 1657–1662, dec 1995.

[71] Fan-Tien Cheng, Jeng-Shi Chen, and Fan-Chu Kung, “Study and resolution of singularities for a 7-DOF redundant manipulator,” *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 469–480, jun 1998.

[72] S. Huang, Y. Peng, W. Wei, and J. Xiang, “Clamping weighted least-norm method for the manipulator kinematic control: Avoiding joint limits,” in *Proceedings of the 33rd Chinese Control Conference*. IEEE, jul 2014, pp. 8309–8314.

[73] S. H. Hyon, J. G. Hale, and G. Cheng, “Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 884–898, 2007.

[74] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, “Towards versatile legged robots through active impedance control,” *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 2015.

[75] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur, “Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations,” in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, nov 2015, pp. 684–689.

[76] B. Siciliano, “Kinematic control of redundant robot manipulators: A tutorial,” *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 201–212, 1990.

[77] M. J. Sung, Young Whee and Cho, Dong Kwon and Chung, “A Constrained

Optimization Approach to Resolving Manipulator Redundancy," *Journal of Field Robotics*, vol. 13, no. 5, pp. 275–288, 1996.

[78] T. Chan and R. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators," in *Proceedings IEEE International Conference on Robotics and Automation.* IEEE Comput. Soc. Press, 1993, pp. 395–402.

[79] F. Flacco, A. De Luca, and O. Khatib, "Control of Redundant Robots Under Hard Joint Constraints: Saturation in the Null Space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, jun 2015.

[80] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2014.

[81] J. Xu, W. Wang, and Y. Sun, "Two optimization algorithms for solving robotics inverse kinematics with redundancy," *Journal of Control Theory and Applications*, vol. 8, no. 2, pp. 166–175, 2010.

[82] L. Sentis, "Compliant control of whole-body multi-contact behaviors in humanoid robots," *Motion Planning for Humanoid Robots*, vol. 26, no. 3, pp. 29–66, 2010.

[83] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[84] N. Mansard and F. Chaumette, "Directional Redundancy for Robot Control," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1179–1192, jun 2009.

[85] M. Nikolic, S. Savic, B. Borovac, and M. Rakovic, "Task prioritization framework for kinesthetic teaching of a free-standing humanoid robot," in *IEEE*

13th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, sep 2015, pp. 241–246.

[86] A. Ben-Israel and T. N. E. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003.

[87] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2939–2944.

[88] A. Escande, N. Mansard, P.-b. Wieber, A. Escande, N. Mansard, P.-b. W. Hierarchical, and Q. Programming, "Hierarchical Quadratic Programming : Companion report," *International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[89] V. Kumar and K. Waldron, "Suboptimal algorithms for force distribution in multifingered grippers," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 491–498, 1989.

[90] P. Gorce, "Dynamic postural control method for biped in unknown environment," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 29, no. 6, pp. 616–626, 1999.

[91] K. Waldron, "Force and motion management in legged locomotion," *IEEE Journal on Robotics and Automation*, vol. 2, no. 4, pp. 214–220, dec 1986.

[92] C. A. Klein and S. Kittivatcharapong, "Optimal Force Distribution for the Legs of a Walking Machine with Friction Cone Constraints," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 73–85, 1990.

[93] Y. Zhang and D. Guo, "Linear programming versus quadratic programming in robots' repetitive redundancy resolution: A chattering phenomenon investi-

gation," in *4th IEEE Conference on Industrial Electronics and Applications*. IEEE, may 2009, pp. 2822–2827.

[94] K. L. Doty, C. Melchiorri, and C. Bonivento, "A Theory of Generalized Inverses Applied to Robotics," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.

[95] F. T. Cheng, T. H. Chen, and Y. Y. Sun, "Resolving Manipulator Redundancy Under Inequality Constraints," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 1, pp. 65–71, 1994.

[96] G. Strang, *Introduction to Linear Algebra*, 4th ed. Wellesley-Cambridge Press Wellesley, MA, 2009.

[97] P.-B. Escande, Adrien and Mansard, Nicolas and Wieber, "Fast Resolution of Hierarchized Inverse Kinematics with Inequality Constraints," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2010, pp. 3733–3738.

[98] N. Mansard, "A dedicated solver for fast operational-space inverse dynamics," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012, pp. 4943–4949.

[99] Youngjin Choi, Doik Kim, Yonghwan Oh, and Bum-Jae You, "Posture/Walking Control for Humanoid Robot Based on Kinematic Resolution of CoM Jacobian With Embedded Motion," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1285–1293, dec 2007.

[100] L. Righetti, M. Mistry, J. Buchli, and S. Schaal, "Inverse Dynamics Control of Floating-Base Robots With External Contraints: an Unified View," in *IEEE International Conference on Robotics and Automation (Icra)*, 2011, pp. 1085–1090.

[101] L. Righetti and S. Schaal, "Quadratic programming for inverse dynamics with optimal distribution of contact forces," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 538–543, 2012.

[102] F. Aghili, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 834–849, 2005.

[103] L. Fraser, B. Rekabdar, M. Nicolescu, M. Nicolescu, D. Feil-Seifer, and G. Bebis, "A compact task representation for hierarchical robot control," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, nov 2016, pp. 697–704.

[104] A. Shkolnik and R. Tedrake, "Inverse kinematics for a point-foot quadruped robot with dynamic redundancy resolution," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4331–4336.

[105] Q. Huang, "Sliding Mode Control Based on Virtual Suspension Model for Controlling Posture and Vibration of Six-Legged Walking Robot," in *Robotics and Biomimetics. ROBIO'06. IEEE International Conference on*, 2006, pp. 642–647.

[106] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, 2009, pp. 814–820.

[107] T. M. Abdel-Rahman, "A Generalized Practical Method for Analytic Solution of the Constrained Inverse Kinematics Problem of Redundant Manipulators," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 382–395, 1991.

[108] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Learning, planning, and control for quadruped locomotion over challenging terrain,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[109] S. Aoyagi, T. Matsuda, T.-W. Kong, T. Ishimaru, M. Suzuki, and K. Inoue, “Proposal and Development of Arrayed Sole Sensor for Legged Robot and Contact Force Detection Using Neural Networks,” *IEEE Sensors Journal*, vol. 11, no. 9, pp. 2048–2056, sep 2011.

[110] X. Xiong, F. Worgotter, and P. Manoonpong, “Adaptive and Energy Efficient Walking in a Hexapod Robot Under Neuromechanical Control and Sensorimotor Learning,” *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2521–2534, nov 2016.

[111] R. D. Beer, H. J. Chiel, R. D. Quinn, K. S. Espenschied, and P. Larsson, “A Distributed Neural Network Architecture for Hexapod Robot Locomotion,” *Neural Computation*, vol. 4, no. 3, pp. 356–365, may 1992.

[112] G. Parker and R. Zbeda, “Learning Area Coverage for a Self-Sufficient Hexapod Robot Using a Cyclic Genetic Algorithm,” *IEEE Systems Journal*, vol. 8, no. 3, pp. 778–790, sep 2014.

[113] Pei-Chun Lin, H. Komsuoglu, and D. Koditschek, “Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 932–943, oct 2006.

[114] ——, “A leg configuration measurement system for full-body pose estimates in a hexapod robot,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 411–422, jun 2005.

[115] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, “Optimal distribution of contact forces with inverse-dynamics control,” *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.

[116] A. Del Prete, N. Mansard, F. Nori, G. Metta, and L. Natale, "Partial Force Control of Constrained Floating-Base Robots," in *EEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3227–3232.

[117] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.

[118] B. Henze, A. Dietrich, and C. Ott, "An Approach to Combine Balancing with Hierarchical Whole-Body Control for Legged Humanoid Robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 700–707, jul 2016.

[119] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 5148–5154, 2015.

[120] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, nov 2014, pp. 295–302.

[121] H. Uchida, "Trajectory Tracking Control for Six-legged Robot by Correction of Leg Link Target Trajectories Based on LQI Control," vol. 6, no. 1, pp. 1–11, 2016.

[122] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson, "Optimal Paths for Landmark-Based Navigation by Differential-Drive Vehicles With Field-of-View Constraints," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 47–59, 2007.

[123] H. Deng, G. Xin, G. Zhong, and M. Mistry, "Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications," *Robotics and Autonomous Systems*, vol. 95, pp. 13–24, 2017.

[124] J. J. Craig, *Introduction to robotics: mechanics and control.* Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005.

[125] A. Robotics and R. N. Jazar, *Theory of applied robotics: kinematics, dynamics, and control.* Springer Science \& Business Media, 2010.

[126] A. S. Deo and I. D. Walker, "Robot subtask performance with singularity robustness using optimal damped least-squares," in *Robotics and Automation. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 434–441.

[127] L. Saab, O. Ramos, N. Mansard, P. Souères, and J. Y. Fourquet, "Generic dynamic motion generation with multiple unilateral constraints," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4127–4133, 2011.

[128] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[129] R. Tinos and M. H. Terra, "Control of cooperative manipulators with passive joints," in *Proceedings of the American Control Conference, Vols 1-6*, 2002, pp. 1129–1134.

[130] R. V. Balafoutis, Constantinos A and Patel, *Dynamic analysis of robot manipulators: A Cartesian tensor approach.* Springer Science \& Business Media, 1991.

[131] A. A. Shabana, *Computational dynamics.* John Wiley \& Sons, 2009.

[132] J. Crane III, Carl D and Duffy, *Kinematic analysis of robot manipulators.* Cambridge University Press, 2008.

[133] K. Harada, E. Yoshida, and K. Yokoi, *Motion planning for humanoid robots.* Springer, 2010.

[134] Z. Tang, Z. Sun, C. Zhou, and L. Hu, "Reference trajectory generation for 3-Dimensional walking of a humanoid robot," *Tsinghua Science and Technology*, vol. 12, no. 5, pp. 577–584, oct 2007.

[135] L. Sciavicco and B. Siciliano, "A dynamic solution to the inverse kinematic problem for redundant manipulators," *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, no. 5, pp. 1081–1087, 1987.

[136] L. Sentis and O. Khatib, "A Whole-Body Control Framework for Humanoids Operating in Human Environments, IEEE, 2006.pdf," in *Robotics and Automation, ICRA. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2641–2648.

[137] M. and others Spong, Mark W and Hutchinson, Seth and Vidyasagar, *Robot modeling and control.* Wiley New York, 2006.

[138] E. Khalil, Wisama and Dombre, *Modeling, identification and control of robots.* Butterworth-Heinemann, 2004.

[139] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "Optimal Force Distribution for the Legs of a Quadruped Robot," *Machine Intelligence & Robotic Control*, vol. 1, no. 2, pp. 87–94, 1999.

[140] C. A. Klein and T. S. Chung, "Force Interaction and Allocation for the Legs of a Walking Vehicle," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 546–555, 1987.

[141] M. Nahon and J. Angeles, "Real-time force optimization in parallel kinematic chains under inequality constraints," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp. 439–450, 1992.

[142] F. T. Cheng and D. E. Orin, "Efficient Formulation of the Force Distribution Equations for Simple Closed-Chain Robotic Mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 1, pp. 25–32, 1991.

[143] S. S. Roy and D. K. Pratihar, “Effects of turning gait parameters on energy consumption and stability of a six-legged walking robot,” *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 72–82, 2012.

[144] F. Aghili, “Quadratically constrained quadratic-programming based control of legged robots subject to nonlinear friction cone and switching constraints,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2469–2479, 2017.

[145] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, “Parallel Elite Genetic Algorithm and Its Application to Global Path Planning for Autonomous Robot Navigation,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4813–4821, oct 2011.

[146] B.-Y. Kang, J. Lee, and D.-W. Kim, “Fast genetic algorithm for robot path planning,” *Electronics Letters*, vol. 49, no. 23, pp. 1449–1451, nov 2013.

[147] J. Wang, J. Wen, W. Chen, H. Yue, and D. Liu, “A gait generating algorithm with smooth speed transition for the locomotion of legged robots,” *Transactions of the Institute of Measurement and Control*, vol. 36, no. 2, pp. 260–275, 2013.

[148] R. E. Udwadia, Firdaus E and Kalaba, “On the foundations of analytical dynamics,” *International Journal of non-linear mechanics*, vol. 37, no. 6, pp. 1079–1090, 2002.

[149] M.-H. Perng and L. Hsiao, “Inverse Kinematic Solutions for a Fully Parallel Robot with Singularity Robustness,” *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 575–583, 1999.

[150] O. Kanoun, F. Lamiraux, and P.-B. B. Wieber, “Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[151] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[152] O. Siciliano, Bruno and Khatib, *Springer handbook of robotics.* Springer Science \& Business Media, 2008.

[153] V. Lumelsky, "Effect of kinematics on motion planning for planar robot arms moving amidst unknown obstacles," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 207–223, jun 1987.

[154] S. Voliotis, G. Panopoulos, and M. Christodoulou, "Co-ordinated path tracking by two robot arms with a noninverting algorithm based on the simplex method," *IEE Proceedings D Control Theory and Applications*, vol. 137, no. 6, pp. 390–396, 1990.

[155] Y. Mezouar and F. Chaumette, "Optimal Camera Trajectory with Image-Based Control," *The International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 781–803, 2003.

[156] Holmberg Robert and Khatib Oussama, "Development and control of a holonomic mobile robot for mobile manipulation tasks," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 1066–1074, 2000.

[157] L. S. A and G. Andrew, "Neural-network control of mobile manipulators," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1121–1133, 2001.

[158] Andy Smith, "Generic SOPs documents," 2016.

[159] J. Florez and C. Parra, "Review of sensors used in robotics for humanitarian demining application," in *IEEE Colombian Conference on Robotics and Automation (CCRA).* IEEE, 2016, pp. 1–6.

[160] H. Najjaran and A. Goldenberg, “Real-time motion planning of an autonomous mobile manipulator using a fuzzy adaptive Kalman filter,” *Robotics and Autonomous Systems*, vol. 55, no. 2, pp. 96–106, 2007.

[161] M. K. Habib, “Humanitarian demining: Reality and the challenge of technology-the state of the arts,” *International Journal of Advanced Robotic Systems*, vol. 4, no. 2, pp. 151–172, 2007.

[162] H. Najjaran and A. a. Goldenberg, “Landmine detection using an autonomous terrain-scanning robot,” *Industrial Robot: An International Journal*, vol. 32, no. 3, pp. 240–247, 2005.

[163] M. Gavilanes, Javier and Fernandez, Roemi and Montes, Hector and Sarria, Javier and de Santos, Pablo Gonzalez and Armada, “Instrumented Scanning Manipulator for Landmines Detection Tasks,” in *Proceedings IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015*, 2015, pp. 180–185.

[164] J. Borgstadt and N. Ferrier, “Visual servoing: path interpolation by homography decomposition,” in *Proceedings ICRA. IEEE International Conference on Robotics and Automation.* IEEE, 2001, pp. 723–730.

[165] Y. Mezouar and F. Chaumette, “Path planning for robust image-based control,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.

[166] B. J. Stephens and C. G. Atkeson, “Dynamic Balance Force Control for compliant humanoid robots,” in *EEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, oct 2010, pp. 1248–1255.

[167] A. A. Zobova, T. Habra, N. Van der Noot, H. Dallali, N. G. Tsagarakis, P. Fisette, and R. Ronsse, “Multi-physics modelling of a compliant humanoid robot,” *Multibody System Dynamics*, vol. 39, no. 1-2, pp. 95–114, 2017.

[168] K. Kojima, Y. Ishiguro, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Rotational Sliding Motion Generation for Humanoid Robot by Force Distribution in Each Contact Face," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2088–2095, oct 2017.

[169] O. Kanoun, "Real-Time Prioritized Kinematic Control under Inequality Constraints for Redundant Manipulators," *Robotics: Science and Systems*, vol. 7, pp. 145–152, 2012.

[170] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[171] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics*, vol. 29, no. 4, p. 1, 2010.

[172] D. Khudher, "Normal walking without arm - YouTube." [Online]. Available: https://www.youtube.com/watch?v=mVRN6DEG3bU{&}feature=youtu.be

[173] ——, "Scanning case of no obstacle - YouTube." [Online]. Available: https://www.youtube.com/watch?v=sxJpPRuoh9o{&}feature=youtu.be

[174] ——, "Turning to left. - YouTube." [Online]. Available: https://www.youtube.com/watch?v=aoDjDELp{_}tU{&}feature=youtu.be

[175] ——, "Scanning with arm case of Slippage - YouTube." [Online]. Available: https://www.youtube.com/watch?v=tBPaHjCY8Bg{&}feature=youtu.be

APPENDIX A

Equation of Motion

$$L = K - U \tag{A.1}$$

where L is Lagrangian, K is kinetic energy, and U is potential energy.

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{q}}) - \frac{\partial K}{\partial \dot{q}} + \frac{\partial U}{\partial q} = \tau \tag{A.2}$$

$$\frac{\partial K}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}}\left[\frac{1}{2}\dot{q}^T M(q)\dot{q}\right] \longrightarrow K = \frac{1}{2}\dot{q}^T M(q)\dot{q} \tag{A.3}$$

$$K = \frac{1}{2}mv^2 \; and \; K = \frac{1}{2}\omega^T I_c \omega \tag{A.4}$$

$$K_i = \frac{1}{2}(v_{c_i}^T m_i v_{c_i} + \omega_i^T I_{c_i} \omega_i) \tag{A.5}$$

Where I_{c_i} is the inertia at CoM.

$$\frac{1}{2}\dot{q}M\dot{q} \equiv \frac{1}{2}(v_{c_i}^T m_i v_{c_i} + \omega_i^T I_{c_i}\omega_i) \tag{A.6}$$

Since $v_{C_i} = J_{v_i}\dot{q}$ and $\omega_{c_i} = J_{c_i}\dot{q}$ then

$$\frac{1}{2}\dot{q}M\dot{q} = \frac{1}{2}\sum_{i=1}^{n}(J_{v_i}^T\dot{q}^T m_i J_{v_i}\dot{q} + J_{\omega_i}^T\dot{q}^T I_{c_i}J_{\omega_i}\dot{q}) \tag{A.7}$$

Since $M = \sum_{i=1}^{n}(J_{v_i}^T m_i J_{v_i} + J_{\omega_i}^T I_{c_i} J_{\omega_i})$

The equation of motion of any leg can be set in a matrix form

$$M(q)\ddot{q} + H(q,\dot{q}) + G(q) = \tau \tag{A.8}$$

Or in summation form

$$\sum_{j=1}^{3} M_{ij}(q)\ddot{q_j} + \sum_{k=1}^{3}\sum_{m=1}^{3} H_{ikm}\dot{q_k}\dot{q_m} + G_i = \tau_i \tag{A.9}$$

$M(q)$ is an 3×3 inertia type symmetric matrix (mas matrix)

$$M = \sum_{i=1}^{3}\left(J_{v_i}^T m_i J_{v[i]} + \frac{1}{2}J_{v_i}^T I_i^0 J_{\omega_i}\right) \tag{A.10}$$

Where J_{v_i} is a 3×3 liner velocity Jacobian J_{ω_i} is $3 \times 3$the angular velocity Jacobain and I_i^0 is the inertia matrix of link i about its CoM and expresses in the base (inertia tensor).

H_{ikm} is Coriolis and centrifugal force matrix (the velocity coupiling vector)

$$H_{ikm} = \sum_{j=1}^{3}\sum_{k=1}^{3}\left(\frac{\partial M_{ij}}{\partial q_{ij}} - \frac{1}{2}\frac{\partial M_{jk}}{\partial q_i}\right) \tag{A.11}$$

And G_i is the gravitational vector

$$G_i = \sum_{j}^{3} m_j g^T J_{v_i} J_{v_i} \tag{A.12}$$

The dynamic equation for three joint leg is written as follows:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{A.13}$$

Appendix B

Equilibrium Equations

$$\sum_{i=1,4,5} f_{ix} + F_x = 0 \tag{B.1}$$

$$\sum_{i=1,4,5} f_{iy} + F_y = 0 \tag{B.2}$$

$$\sum_{i=1,4,5} f_{iz} + F_z = 0 \tag{B.3}$$

$$\sum_{i=1,4,5} y_i f_{iz} - \sum_{i=1,4,5} z_i f_{iy} + y_c F_z - z_c F_y + M_x = 0 \tag{B.4}$$

$$\sum_{i=1,4,5} z_i f_{ix} - \sum_{i=1,4,5} x_i f_{iz} + z_c F_x - x_c F_z + M_y = 0 \tag{B.5}$$

$$\sum_{i=1,4,5} x_i f_{iy} - \sum_{i=1,4,5} y_i f_{ix} + x_c F_y - y_c F_x + M_z = 0 \tag{B.6}$$

Where $F_i = [f_{ix}, f_{iy}, f_{iz}]$ are the ground reaction force on foot i and $i = 1, 4, 5$ (legs in support). $W = [F_x, F_y, F_Z, M_x, M_y, M_z]^T$ are the wrench (Force and the moments) acting at the robot CoM and represent the robot payload. $[x_i, y_i, z_i]$ are the coordinate of lag i.

These equations are normally written in a matrix form as follows:

$$A_{1,4,5}.F_{1,4,5} = (-BW) \tag{B.7}$$

Appendix C

Simulink Environment of the Robot

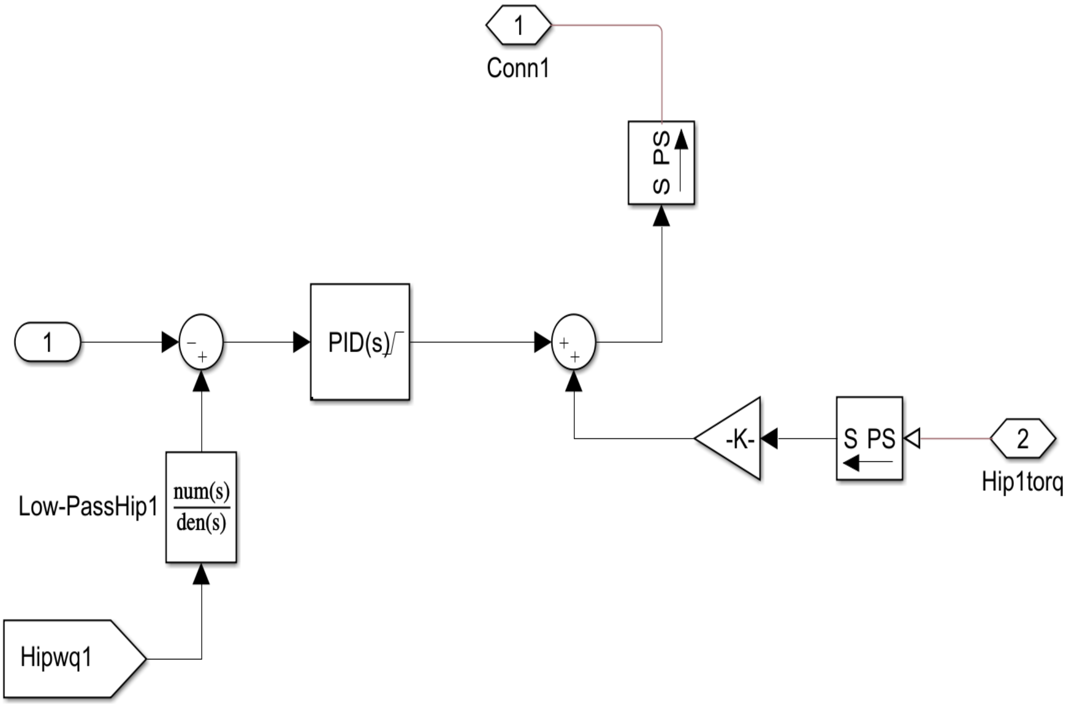


Figure C.1: Simulink block diagram for PID controller of the robot.

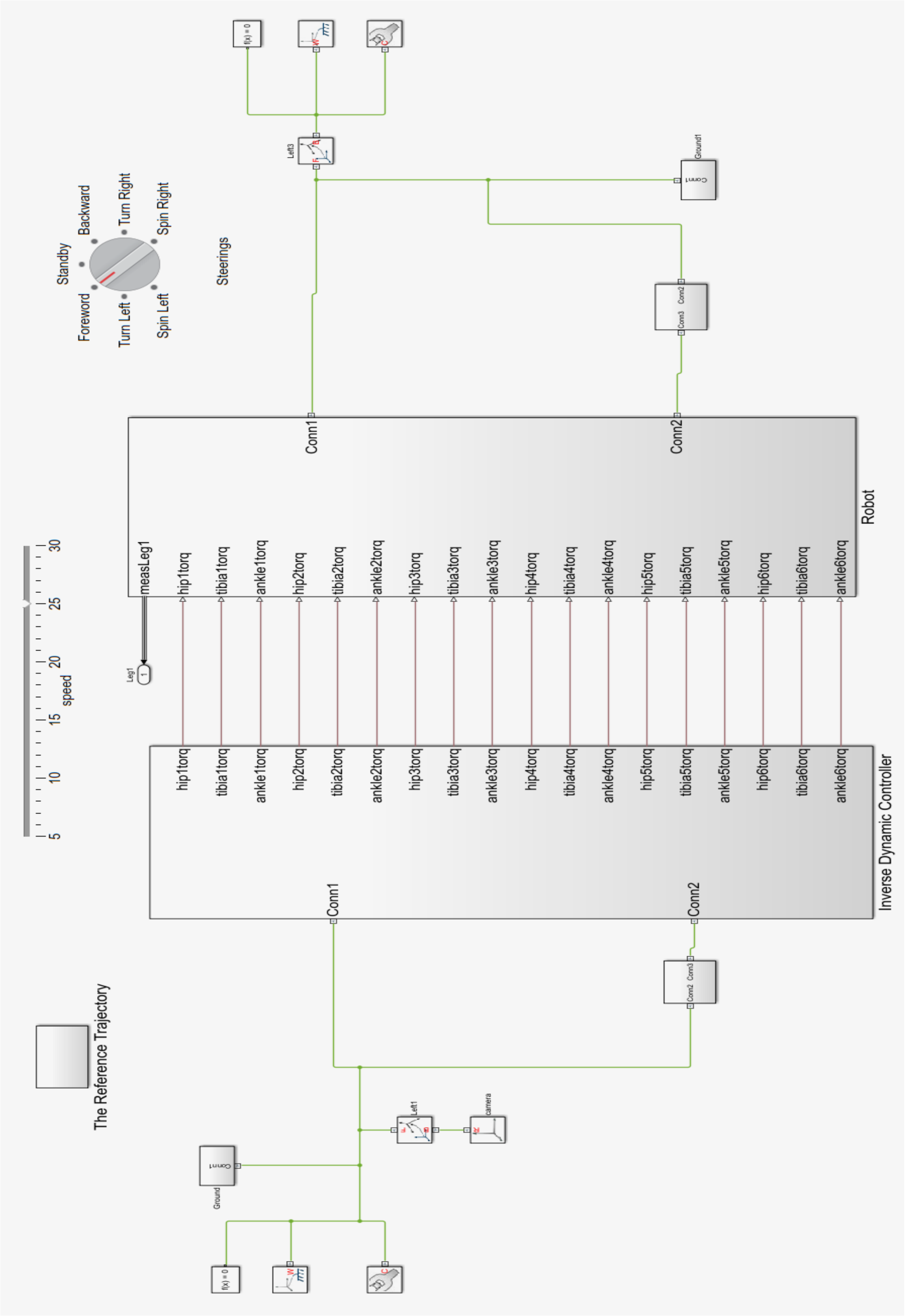


Figure C.2: Simulink block diagram for inverse dynamics controller of the robot.

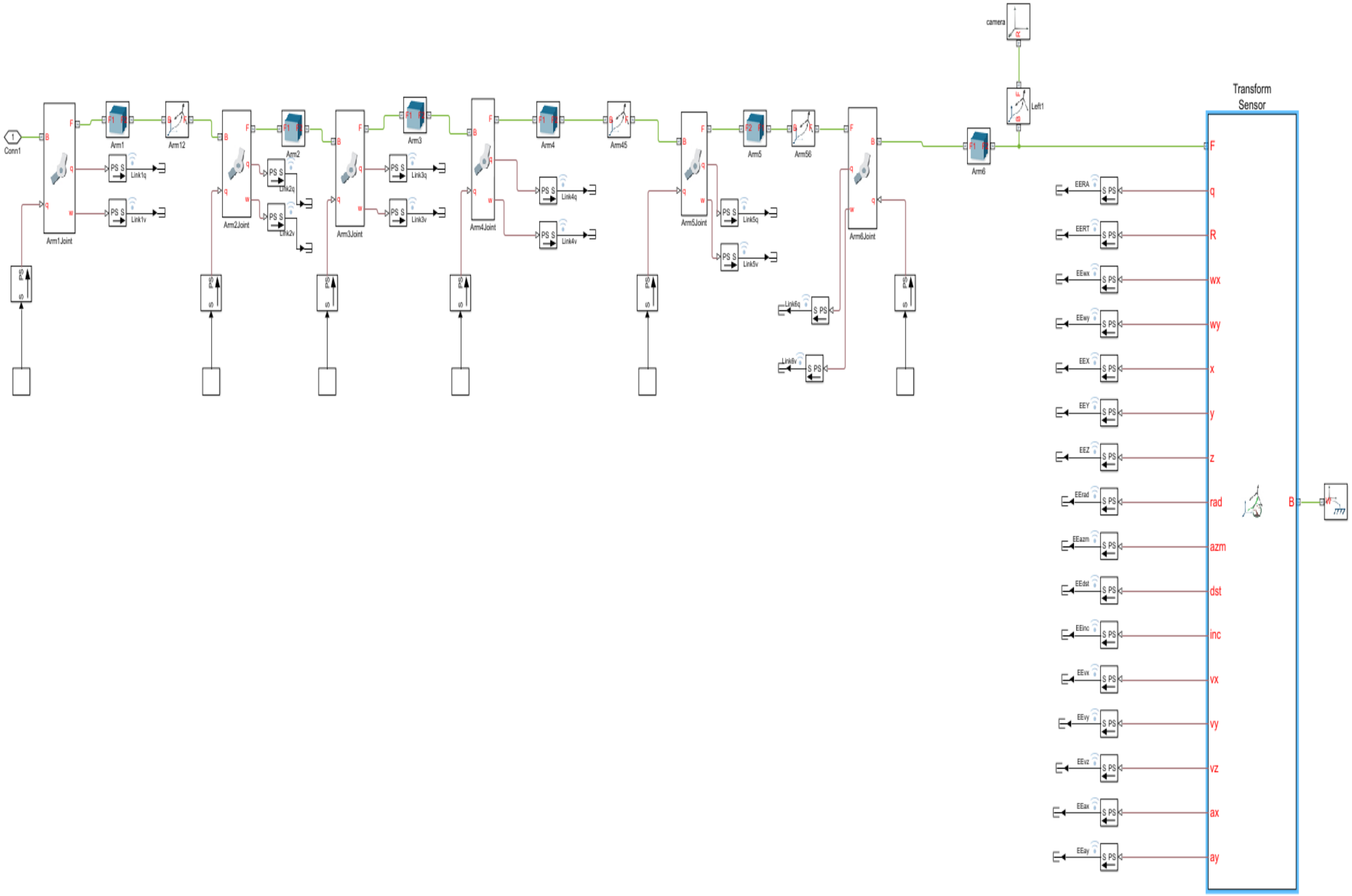


Figure C.3: Simulink block diagram for the robot arm.